

Topics in Educational Cyber-Physical Labs: Configurations, Data Collection and Analysis

THÈSE N° 8471 (2018)

PRÉSENTÉE LE 27 AOÛT 2018

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR

GROUPE SCI STI DG

PROGRAMME DOCTORAL EN GÉNIE ÉLECTRIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Wissam HALIMI

acceptée sur proposition du jury:

Prof. P. Frossard, président du jury
Dr D. Gillet, Dr C. Salzmann, directeurs de thèse
Prof. L. de la Torre Cubillo, rapporteur
Prof. F. Moghaddam, rapporteuse
Prof. F. Mondada, rapporteur



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2018

“[...] remember why you started”

— Chris Burkmenn



Acknowledgement

My journey here started in the least expected of ways. I had drafted an email to apply for an internship. I had started writing it, and then I wasn't sure if I should apply. I saved the draft "for later". A couple of weeks later, a Sandy Ingram emails me, asking me for further info to consider my application. Turns out I had sent that incomplete email with my CV attached! I wasn't sure what to do. Well, I sent the needed info... And here I am.

Denis (a.k.a the PhD Advisor), thank you for opening the doors for me, for providing me with the opportunity to be here, to participate in challenging multidisciplinary projects and be part of something interesting. Christophe (a.k.a the PhD Co-Advisor), thank you for taking me in, for the long discussion hours, and for walking through a thought path when it was needed.

REACT as I know it, thank you! There I met very dear people. Sandy, thank you for being a mentor and a good friend! Thank you for the open-ended discussions, quick Q&As, and for being kind and caring, for being you. Laleh and Li Na, thank you for coffee breaks, tea breaks, walks, talks and everything-related. Thank you for being by side without me asking, for your support and for being more than friends. Things wouldn't be the same without you!

Andrii, I have 2 words for you: bread and banana! Thank you for being a listening ear, a great office mate, for the time you dedicated for our discussions and you're sincere interest in landing a hand. Ezequiel and Juan Carlos I will miss coming to your office to chat and pick on you. Andre and Alex it was great knowing and working amongst you.

The Lebanese Gang and co. (a.k.a "PSF"): Christopher, Sabine, Maroun and Elodie. Thank you for being uncomplicated people and a piece of home away from home. Lunch breaks, "quick afternoon breaks" will never be the same. We are the loudest bunch wherever we go, and I am sure we make heads turn (mainly because of our noise). I feel like a kid around you.

To my family, thank you for raising me the best you knew how. To my in-laws, thank you for welcoming me into your family 6 years ago. Thank you both for being supportive and enthusiastic about my career choices. Deeply appreciated.

Ahmad, my life companion, you are the surest thing. Being with you has been so easy, no questions asked. Thank you for being such an understanding, tolerating and encouraging person. Can't wait for what's to come!

Lausanne, 8 July 2018

W. H.



Funding Acknowledgements

This thesis was partially funded by the European Union in the context of the Go-Lab project (grant agreement no. 317601) under the ICT theme of the 7th Framework Programme for R&D (FP7), the Next-Lab project (grant agreement no. 731685) Horizon 2020 Research and Innovation Program, and the Swiss National Science Foundation in the framework of its SCOPES (Scientific co-operation between Eastern Europe and Switzerland) program (grant agreement no. IZ74Z0-160454).



Abstract

Recent advances in remote sensing and actuation technologies, coupled with the large reach of the internet, allowed for the emergence of applications such as cyber-physical labs. Cyber-physical labs are the digital and remotely-accessible equivalent of the lab equipment students use in school to experiment, through web-based interfaces. Students as the intended users, teachers as the educational content curators and lab owners as the service providers derive value from these systems, they are our stakeholders.

In this thesis, we take a close look at issues pertaining to the development and deployment of cyber-physical labs in educational settings, and propose new approaches to address them. Moreover, we study the use of such systems in real-settings, to infer the impact of the experimental behavior of students on their academic performance.

First, we tackle the case of the automatic generation of web apps interfacing cyber-physical labs. It is the equivalent of preparing experiments by teachers through arranging the equipment for multiple experiments with the same setup. We propose an extension to the Smart Device Specification for cyber-physical labs, and a tool which automatically generates these apps based on it. The generated apps implement the necessary functions to fully use a cyber-physical lab, and are ready to be integrated in online learning platforms.

Next, we investigate issues related to the collection and retrieval of students' generated data through their interaction with cyber-physical labs. We elicit the requirements for an activity tracking infrastructure next to students and lab owners. We accordingly propose an activity tracking infrastructure which is based on two components: a vocabulary and an architectural model. The proposed vocabulary ensures deriving adequate insights from the recorded activity, and the proposed architecture addresses privacy and data access issues pertaining to students and lab owners respectively. We evaluate our proposal with two example cyber-physical labs.

Last, we collect the interaction data of a cyber-physical lab used in a MOOC. We devise computational analyses on the students activity statistics, in search for indicators of academic performance. We find that high and low performing students show some statistically different activity statistics. Then, we sequence the actions students did in experiments, and don't find any statistically significant patterns for low and high-performing students. The analyses provide insights on the usage of installed facilities to service a potential massive access of users to limited resources (lab installations), and shed light on possible indicators of academic performance.

Keywords: Cyber-physical Labs, remote Labs, digital education, learning analytics, educational data mining, MOOC

Résumé

Les progrès récents dans les technologies de télédétection et télécommande, associées à la large portée d'Internet, ont permis l'émergence d'applications telles que les laboratoires cyber-physiques. Les laboratoires cyber-physiques sont l'équivalent numérique et accessible à distance de l'équipement de laboratoire que les élèves utilisent à l'école pour expérimenter, au moyen d'interfaces Web telles que les applications Web. Les étudiants, les enseignants et les propriétaires de laboratoire tirent de la valeur de ces systèmes, ils sont nos parties prenantes. Les étudiants sont les utilisateurs, les enseignants sont les fournisseurs de contenu éducatif et les propriétaires de laboratoire sont les fournisseurs de services.

Dans cette thèse, nous examinons de près les questions relatives aux laboratoires cyber-physiques et nous proposons de nouvelles approches pour y remédier. Nous analysons également l'utilisation de tels systèmes dans un MOOC, afin de détecter l'impact du comportement expérimental des étudiants sur leurs performances académiques.

D'abord, nous abordons le challenge de la génération d'applications Web qui interfacent les laboratoires cyber-physiques. Pour les enseignants, c'est l'équivalent de préparer des expériences en organisant les composant du laboratoire pour de multiples expériences avec le même équipement. Nous proposons une extension de la spécification Smart Device pour les laboratoires cyber-physiques, et un outil qui génère ces applications en fonction de celui-ci. Les applications générées automatiquement implémentent les fonctions nécessaires à l'utilisation d'un laboratoire cyber-physique, et sont prêtes à être intégrées dans les plates-formes d'apprentissage en ligne.

Ensuite, nous étudions les problèmes liés à la collecte et à la récupération des données générées par les étudiants suite à leur interaction avec les laboratoires cyber-physiques. Nous considérons les besoins des étudiants et des propriétaires de laboratoires. Nous sollicitons les exigences d'une infrastructure de suivi des activités composée d'un vocabulaire et d'un modèle architectural auprès des étudiants et propriétaires de laboratoires. Le vocabulaire proposé garantit la valorisation de l'activité enregistrée, et l'architecture proposée traite respectivement des problèmes de confidentialité et d'accès aux données des étudiants et des propriétaires de laboratoire. Nous évaluons notre proposition avec deux exemples de laboratoires cyber-physiques.

Enfin, nous collectons les données d'interaction avec un laboratoire cyber-physique utilisé dans un MOOC. Nous concevons des analyses computationnelles sur les statistiques d'activité des étudiants, à la recherche d'indicateurs de performance académique. Nous constatons que les élèves les plus performants et les moins performants présentent des caractéristiques d'activité statistiquement différentes. Ensuite, nous séquençons les étapes que les élèves ont faites dans une expérience, et nous ne trouvons pas de modèles statistiquement significatifs pour les étudiants

à faible et haute performance. Cette analyse donne un aperçu de l'utilisation des installations installées pour desservir un accès potentiellement massif à des ressources limitées (installations de laboratoire) et fait la lumière sur les indicateurs possibles de performance académique.

Mots clefs : laboratoires cyber-physiques, laboratoires à distance, exploration de données éducatives, MOOC

Contents

Acknowledgement	i
Funding Acknowledgements	iii
Abstract (English/Français)	v
List of figures	xiii
List of tables	xvii
1 Introduction	1
1.1 Cyber-Physical Labs	2
1.2 Educational Web-based Platforms	4
1.2.1 The Go-Lab Infrastructure	4
1.2.2 MOOC Platforms	8
1.3 Data collection in Digital Educational Settings	9
1.3.1 Interaction Continuity	9
1.3.2 Activity Tracking	11
1.4 Educational Data Analysis	13
1.5 CPLs as Modular Open Educational Resources	14
1.5.1 Background	14
1.5.2 Formalizing the Integration of CPLs in Educational Web-Based Platforms	15
1.5.3 Open Educational Labs	15
1.5.4 Proof-of-concept Implementation	17
1.6 Running Examples	19
1.6.1 The Mach-Zehnder Interferometer	19
1.6.2 The Electrical Servo Drive	21
1.7 Challenges, Contributions and Validation	22
1.7.1 Enabling the Automatic Generation of Web-Apps for CPLs	22
1.7.2 Activity Tracking Infrastructure for CPLs	23
1.7.3 Understanding How Students Access and Use CPLs	24
1.8 Thesis Outline	25

2	The Smart Device Configurations	27
2.1	Related Work	28
2.2	The Smart Device Paradigm and Specification	30
2.3	The Smart Device Configurations	31
2.4	Validating and Evaluating the Smart Device Configurations with the Automatic Web App Generator	35
2.4.1	Integration in LTI-Consumer Platforms	39
2.4.2	Integration in Graasp and Interoperability with the GoLabz Infrastructure	40
2.5	Proof-of-concept Example	40
2.6	Discussion	43
2.7	Conclusion	44
3	Activity Tracking Infrastructure for Embedded Cyber-Physical Labs	45
3.1	Related Work	46
3.1.1	Activity Tracking Architectures in Online Learning Environments	46
3.1.2	Activity Tracking Data Formats in Online Learning Environments	47
3.1.3	Privacy in Online Learning Environments	48
3.2	Requirements Elicitation	49
3.2.1	Students Questionnaires	49
3.2.2	Lab Owners Questionnaire	60
3.3	Infrastructure Requirements	70
3.4	Infrastructure Vocabulary	71
3.4.1	The xAPI Specification Explained	72
3.4.2	Extending the xAPI Profiles and Recipes for CPL Experiences	75
3.5	Infrastructure Model	77
3.6	Implementation Examples	81
3.6.1	Tracking Students' Activity on the Servo Motor - A User App Implementation	81
3.6.2	Tracking Students' Activity on the Mach-Zehnder Interferometer - A Lab Owner's Implementation	86
3.7	Evaluation of the Infrastructure with Students	88
3.8	Discussion	96
3.9	Conclusion	98
4	Computational Analysis of Students' Access and Use of CPLs	99
4.1	Related Work	100
4.2	The Control Systems Lab MOOC	102
4.2.1	Logistics	102
4.2.2	Remote access	103
4.3	Data Collection Framework	104
4.3.1	Duration of Collection and Data	104
4.3.2	Privacy Impact Assessment	105
4.3.3	The Sequences Dataset	107

4.4	Descriptive Statistics of Access to the Servo Drive CPL	107
4.4.1	Location of connection	108
4.4.2	Queue Sizes	109
4.4.3	Concurrent Access	110
4.4.4	Duration of experimentation	110
4.5	Descriptive Statistics of Experimental Behavior	111
4.5.1	Analysis of participation	111
4.5.2	Analysis of activity	113
4.5.3	Discussion	121
4.6	Studying the Effects of Students' Experimental Behavior on Academic Performance	122
4.6.1	Method	122
4.6.2	Results	125
4.6.3	Discussion	126
4.7	Mining Students' Experimental Patterns	127
4.7.1	Method	128
4.7.2	Results	128
4.7.3	Discussion	133
4.8	Summary of Findings and Discussion	134
4.9	Conclusion	135
5	Conclusion and Outlook	137
A	Appendix A: The directory of the xAPI statements for the running examples	141
A.1	xAPI statments for the MZI server-side activity tracking example.	141
A.2	xAPI statments for the Servo Drive Client-App activity tracking example.	145
	Bibliography	155
	List of Abbreviations	157
	Curriculum Vitae	159

List of Figures

1.1	General architecture topology of a CPS with 2 tiers: the computational and environmental tiers.	3
1.2	General architecture topology of CPS applied to cyber-physical labs systems. . .	4
1.3	The landing page of the Golabz platform, where teachers can find and share apps, labs and ILS.	5
1.4	The teacher view of an ILS in Graasp at the Investigation phase, here the teacher used a remote lab and a scaffolding tool	6
1.5	The student view of an ILS at the Investigation phase which shows an embedded remote lab.	7
1.6	The student view of an ILS at the Investigation phase which shows a scaffolding tool (graphing tool).	7
1.7	Control Systems MOOC overview. At the top in the red box, you can see the course's main pages. On the left, you can see the course's modules. The horizontal blue box show the sequence of the selected module.	9
1.8	The Vault space destined for saving and retrieving files, only accessible to the teacher	10
1.9	When the Vault is opened, we can see the contents are generated and used by apps in the ILS, in this case it's a single data file.	10
1.10	The ad-hoc solution for edX: an external database holds the experimental data and serves to the tools [figure adapted from [62]]	11
1.11	An example learning dashboard in Graasp made of two activity visualization apps: (1) the Active Users app on the top which shows in real-time which students are in which phase of the ILS; and (2) the Time Spent app in the bottom which shows the time spent per student in the each phase of the ILS [Source: [76]]. . . .	12
1.12	Example dashboard provided by Open edX showing student engagement on the week of October 15 th , 2017. It shows the number of active learners, videos watched and problems attempted by students.	12
1.13	A screenshot of the data download option for instructors in Open edX where they can choose from a variety of data structures and formats.	13
1.14	Formalization of the integration layers for cyber-physical labs in online learning platforms as modular OERs.	16
1.15	The 3 layers of integration for the Mach-Zehnder interferometer as an OEL. . . .	18
1.16	Basic Mach-Zehnder interferometer layout	20

List of Figures

1.17	Picture of the Mach-Zehnder interferometer.	20
1.18	The electrical servo drive system used by the students of the Control Systems Lab course [annotated from [64]].	21
1.19	Top view of the box where the electric servo drive is lodged [annotated from [64]].	22
1.20	A graphical representation of Figure 1.19 [annotated from [64]].	22
2.1	The lab's web app is disaggregated into mini-apps which can be re-assembled according to a selection of to the lab components they interface. [Source: [59]] .	29
2.2	Requests and data exchange diagram between a CPL implementing the extended Smart Device Specification and the automatic web app generator.	36
2.3	The landing page of the automatic web app generator showing the two available experiment configurations for the Mach-Zehnder lab	37
2.4	The generated cyber-physical lab client application on Graasp for the Mach-Zehnder lab	38
2.5	LTI consumer and provider communication	39
2.6	Mach Zehnder Interferometer layout design by teachers at Gymnase de Morges. .	41
2.7	Mindmap of two possible Mach-Zehnder experiments	41
3.1	Proposed activity tracking model for CPLs supporting the needs of students. . . .	78
3.2	Sequence diagram showing the student granting access to the data storage to save activity traces from the interaction with the CPL web app.	79
3.3	Sequence diagram showing the data flow between the data storage, the awareness and reflection apps, and the graphing apps, when the student has granted access to activity tracking. In this case, all of the apps consume the data from the data storage.	79
3.4	Sequence diagram showing the data flow between the data storage, the awareness and reflection apps, and the graphing apps, when the student has not granted access to activity tracking. In this case, awareness and reflection apps are not functional. The graphing apps consume the data from a supported mechanism through the learning platform.	79
3.5	Proposed activity tracking model for CPLs supporting lab owners	80
3.6	Sequence diagram showing the data flow between the lab owner platform, the data storage, and the lab usage analytics apps, when the lab owner has activated activity tracking and granted access to the analytics apps.	81
3.7	The added tab which leads the learners to the activity tracking configuration page	82
3.8	Activity tracking configuration panel, the students are provided with information regarding the tracking mechanism, where their data is and how it will be used if they activate the tracking.	82
3.9	If students activate tracking, they get access to the dashboard through the panel, and they can deactivate tracking at any time.	83
3.10	Zoomed out overview of the dashboard provided to MOOC students. There are 6 graphs. On the top right is the Deactivate Tracking button.	84

3.11	Example collective graph, showing the number of actions exerted on the CPLs by the whole class, through a defined period of time.	84
3.12	Example individual graph, showing the number of actions exerted on the CPLs by the logged user, through the same defined period of time as in Figure 3.11. . .	85
3.13	Example collective dashboard, showing the total time spent by the whole class, experimenting in Module, grouped by experiments (Module 1-e2 denotes experiment experiment 2 in Module 1).	85
3.14	The landing page of the proposed dashboard app. Lab owners should use their LRS accounts to retrieve the data.	87
3.15	Example of a lab owner visualization in the proposed dashboard app. Lab owners using this app can see the activity on their CPL by day. Each bar represents the number of actions exerted on the CPL.	87
3.16	Example of a lab owner visualization in the proposed dashboard app. Lab owners using this app can see how many actions users did on the CPL. Each bar is unique to a connection/user id.	87
4.1	A typical lesson or module structure with consecutive tabs on the upper horizontal strip, each comprising a different type of learning resources: a video, quiz, CPL web app and other.	102
4.2	The servo drive lab web app user interface composed of a command strip (1), an oscilloscope graph (2), a real-time video feed transmitting the action of the motor (3), and a drop down menu to choose a specific CPL web app from the farm of 25 setups.	104
4.3	Distribution of the origins of connections to the CPLs	108
4.4	Number of connections per day, on the days where the CPLs were accessed. The red bars correspond to the days of the scheduled lab sessions at the university. . .	109
4.5	Time spent on experimentation on the days where the CPLs were accessed through the whole period of the data collection. The red bars correspond to the days of the scheduled lab sessions at the university.	111
4.6	Boxplots showing the distributions of the number of actions exerted per sequence (experimental attempt), for experiments 1 through 15, denoted by e1 through e15 respectively.	114
4.7	Boxplots showing the distributions of the number of actions exerted per sequence (experimental attempt), for experiments 16 through 21, denoted by e16 through e21 respectively.	115
4.8	Boxplots showing the distributions of the time spent in experiments 1 through 15, denoted by e1 through e15 respectively. Only the median of the distribution is shown.	117
4.9	Boxplots showing the distributions of the time spent in experiments 16 through 21, denoted by e16 through e21 respectively. Only the median of the distribution is shown.	118

List of Figures

4.10	Boxplots showing the distributions of the time spent in experiments 1 through 15, denoted by e1 through e15 respectively, after filtering out 0 seconds entries. . . .	119
4.11	Boxplots showing the distributions of the time spent in experiments 16 through 21, denoted by e16 through e21 respectively, after filtering out 0 seconds entries.	120
4.12	Boxplots showing the distributions of the total time spent by low and high-performing students experimenting, throughout the whole duration of the MOOC.	124
4.13	Boxplots showing the distributions of the number of parameter changes for attempts (sequences), for low and high-performing students.	124
4.14	Boxplots showing the distributions of the total number of experimental attempts for low and high-performing students, throughout the whole duration of the MOOC.	125
4.15	Frequent subsequences of actions for e1 (no discrimination between groups) . . .	129
4.16	Frequent subsequences of actions for e2 (no discrimination between groups) . . .	129
4.17	Frequent subsequences of actions for e3 (no discrimination between groups) . . .	129
4.18	Frequent subsequences of actions for e4 (no discrimination between groups) . . .	129
4.19	Frequent subsequences of actions for e5 (no discrimination between groups) . . .	130
4.20	Frequent subsequences of actions for e6 (no discrimination between groups) . . .	130
4.21	Frequent subsequences of actions for e7 (no discrimination between groups) . . .	130
4.22	Frequent subsequences of actions for e8 (no discrimination between groups) . . .	130
4.23	Frequent subsequences of actions for e9 (no discrimination between groups) . . .	130
4.24	Frequent subsequences of actions for e10 (no discrimination between groups) . .	130
4.25	Frequent subsequences of actions for e11 (no discrimination between groups) . .	131
4.26	Frequent subsequences of actions for e12 (no discrimination between groups) . .	131
4.27	Frequent subsequences of actions for e13 (no discrimination between groups) . .	131
4.28	Frequent subsequences of actions for e14 (no discrimination between groups) . .	131
4.29	Frequent subsequences of actions for e15 (no discrimination between groups) . .	131
4.30	Frequent subsequences of actions for e16 (no discrimination between groups) . .	131
4.31	Frequent subsequences of actions for e18 (discrimination between groups) . . .	132
4.32	Frequent subsequences of actions for e17 (no discrimination between groups) . .	132
4.33	Frequent subsequences of actions for e19 (no discrimination between groups) . .	132
4.34	Frequent subsequences of actions for e20 (no discrimination between groups) . .	132
4.35	Frequent subsequences of actions for e21 (no discrimination between groups) . .	132

List of Tables

2.1	Parameters type detected fromt the Smart Device metadata and UI elements assignments	38
4.1	Enrollment in the Automatic Control and Control Systems courses vs. the enrollment in the MOOC	103
4.2	Parameters used in the traces and their descriptions	105
4.3	Parameters encoding table. The symbols are used in the sequences for processing.	107
4.4	Mean, standard deviation, and allocated time in minutes:seconds format, number of experimentation tabs, and allocated duration per experimentation tabs in seconds, for each module.	110
4.5	Participation table showing the average number of times students attempted an experiment.	112
4.6	The number of steps in the instructor-suggested sequences and the allocated time for each experiment.	112
4.7	The p-values for the Kruskal Wallis test for the total duration spent experimenting throughout the course, the number of parameter changes per sequence, the total number of experiment attempts, and class section.	125

1 Introduction

A Cyber-Physical Lab (CPL) is an educational laboratory setup with its instrumentation accessible and controlled through the mediation of computer networks from a distant location. The exploitation of CPLs is affected by the at-the-time available Information and Communication Technologies (ICT) such as communication networks and computation powers. In the 1980s, labs were mainly made of basic physical equipment, which students can use through a tangible mean: being physically present where the lab equipment is installed. Later in the early 1990s, the personal computer emerged and access to peripherals connected to it was possible through screen sharing. Then with the internet spreading beyond 1997 and the boom of associated web technologies in the 2000s, development of tools to gain control over lab equipment at distance became possible. More specifically, with the rise of the web and the internet it was possible to publish static content such as texts and make them available to anyone with a device and a suitable connection. Then, when Web 2.0¹ made a debut, different types of media flooded the web: videos and interactive content such as web applications for chatting and gaming. In parallel, advances in the hardware fields made remote sensing and actuation possible, first based on the development of Radio Frequency Identification technologies (RFID) which allowed the tracking of objects in real time, and now Cyber-Physical Systems (CPS) and the internet of Things (IoT) [43] which support a real time sensing and control of remotely connected sensors and actuators. These are the technologies enabling the development of applications such as Cyber-Physical Labs (CPL) [60] also known as Remote Laboratories (RL). Consequently, the development of cyber-physical labs has enabled live demos during ex cathedra classes, the sharing of expensive lab equipment among institutions, and promoted a 24/7 availability for hands-on sessions carried out by the students online through a web client application.

CPLs emulate the hands-on sessions which are essential for the process of learning and assimilating scientific and engineering concepts [15, 27, 29, 48]. They allow learners to experiment in

¹The second stage of the development of the internet, characterized by the move from static web pages to dynamic or user-generated content and the emergence of social media [52]

order to validate or refute a hypothesis, accept or reject a taught subject. Acquiring and operating laboratory equipment can be a burden on institutions which have limited budget and space for such installations. CPLs are not only a budgetary solution [4], but are a main component of online STEM (Science Technology and Mathematics) and engineering education. In recent years, a number of projects supported the development, deployment and dissemination of cyber-physical labs as educational resources which can be integrated in online learning environments such as educational social media platforms and MOOCs (Massive Open Online Course).

A number of stakeholders derive value from CPLs: **lab owners** as service providers who develop and deploy the systems, **teachers** as content curators for educational resources who personalize and provide the learning content, and **students** as users who exploit those labs for learning. But current system design modalities of cyber-physical lab systems pose a number of problems for their development, deployment, use and interoperability with other applications which hinder their dissemination. In this thesis, we first address topics related to the integration and cooperation of cyber-physical labs in and with educational web based platforms: we enable teachers to automatically generate web applications to access the cyber-physical labs, and propose a platform-independent infrastructure for saving learners' traces when using the cyber-physical labs, taking into the consideration the differing interests of learners and cyber-physical lab providers. And second, we investigate the use of cyber-physical labs in MOOCs (Massive Open Online Courses) to study the effect of how students approach the practice part of a course on academic performance.

This chapter is dedicated to introducing the context of this thesis through first defining Cyber-Physical Labs then describing two main types of web-based educational platforms which are target platforms in this thesis: Graasp² – an educational social media platform and Open edX a MOOC platform. Then, we present two cyber-physical lab examples which are used throughout this text. Last, we provide an overview of the upcoming chapters, challenges and associated contributions.

1.1 Cyber-Physical Labs

Cyber-Physical Labs is a term derived from Cyber-Physical Systems or CPS, which refers to systems that provide access to physical processes through web based interfaces. Example of CPS applications include process control systems, robotics, medical monitoring and others. Cyber-Physical Systems are based on the collaboration between physical and computational processes. The physical counterpart is typically composed of sensors and actuators, which respectively reflect and alter the state of the target environment. The sensors provide data that can be used for purposes such as feedback inputs to the computational processes like automation, or as monitoring data reflecting the state the physical world [44]. The collaboration between the physical and computational entities is ensured mainly by embedded devices, which serve as both communication and interfacing components: the cyber world [83].

²<http://graasp.eu/>

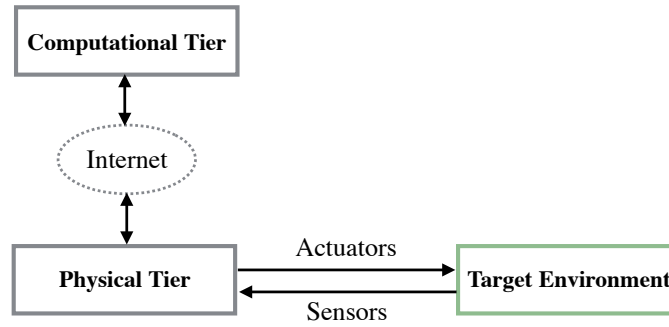


Figure 1.1 – General architecture topology of a CPS with 2 tiers: the computational and environmental tiers.

Given that CPS have a wide diversity of applications, many architectural prototypes exist, each serving best the needs of the respective application (application specific, or vertical/domain specific). In [44] and [69] the authors claim that inherently the architecture of CPS is composed of 2 tiers: the computational (or cyber) and the physical. Figure 1.1 depicts this topology.

In this architecture, the **computational tier** also referred to as the **cyber tier** is responsible for orchestrating the access to the **physical tier**. It provides users with interfaces to interact with the physical system. A user calls the desired services to be executed (for e.g. read from a sensor or set to an actuator). The physical tier receives orders from the computational tier and executes. This tier is solely composed of the sensors and actuators, in addition to any needed hardware to give the physical devices computational and networking capabilities. The physical tier translates and changes the state of the target environment according to decisions made on the cyber level.

Cyber-Physical Labs or remote labs are an application of cyber-physical systems, where the lab's apparatus is the **target environment** being actuated and sensed by a remote user in real-time. Through web interfaces (**computational or cyber tier**), the user connects to the lab and sends out their commands and receives the responses. The lab server (**physical tier**) takes care of requests and responses. Figure 1.2 shows a general architecture for a CPL similar to the general CPS architecture shown in Figure 1.1.

The development, deployment and adoption of cyber-physical labs was and still is hindered by many design considerations. The selection of the software design pattern for CPS applications determines the characteristics of the interfaces to physical devices. For instance, the object-oriented programming (OOP) approach [5, 68] also known as a type of creational design patterns strongly links the application software to the physical devices, by translating specifications to executables dependent on the service provider (in our case the lab owner) and the target system. The service oriented computing (SOC) paradigm exposes a system through well defined descriptors such as APIs ³, which makes application building loosely-decoupled from the service

³An application programming interface is a set of clearly defined “*methods of communication between various software components. A good API makes it easier to develop a computer program by providing all the building blocks,*

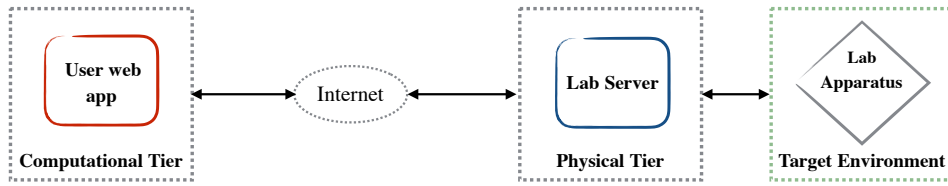


Figure 1.2 – General architecture topology of CPS applied to cyber-physical labs systems.

provider, and hence promoting portability, modularity and interoperability.

1.2 Educational Web-based Platforms

Online learning environments such as Learning Management Systems (LMS), educational social media and MOOC platforms are the digital equivalent of a physical classroom, where teachers come to deliver content and students are the audience. These environments serve at once two purposes: they are the authoring tool for teachers, where they gather the content and structure it; and the interaction medium for students. In this thesis, we are interested in the case where cyber-physical labs are part of a structured learning activity embedded in an educational social media platform and a MOOC platform. In this section, we introduce the framework of the Go-Lab project and its underlying infrastructure, in addition to an institutional (EPFL's) instance of edX⁴, Open edX⁵.

1.2.1 The Go-Lab Infrastructure

Go-Lab is a European funded project under the Seventh ICT Framework Program (FP7). The main goal of Go-Lab is to provide online learning tools which help in spreading scientific knowledge and encouraging students to take up STEM (Science Technology Engineering and Mathematics) majors for their future careers [17, 34, 35, 36]. The Go-Lab project relies on a technical infrastructure which provides online services to teachers and students. Teachers can create online lessons by looking up educational resources on the Go-Lab repository Golabz and elsewhere on the web, and aggregate them in a structured pedagogical scenario in Graasp, before sharing them with students as Inquiry Learning Spaces (ILS).

which are then put together by the programmer” [80] to build a system with a specific functionality through service composition.

⁴<https://www.edx.org/>

⁵Previously <https://open.edx.org/> and now <https://courseware.epfl.ch/>

Golabz Repository

Golabz⁶ is the repository through which teachers can find Open Educational Resources (OER) such as apps, labs, and Inquiry Learning Spaces. Teachers can also share their ILSes on Golabz to be reused by others. The educational resources available on Golabz are scaffolding apps which help teachers and students through a learning activity, online labs such as simulations or cyber-physical labs, or ILSes prepared by other teachers and ready-to-be-used as is, or adapted in Graasp. These resources are curated by the community: teachers and lab providers; and are approved by the Golabz administrators. Golabz is also a platform for lab providers to publish and promote their labs. The Golabz web platform landing page is shown in Figure 1.3.

Graasp

Graasp– the authoring tool and platform embedding the educational resources, provides a medium for teachers to add and organize educational resources using the ILS template. Graasp has a mechanism for integrating third-party applications enabling them to use its proprietary services: context information, user identity, activity tracking, saving and retrieving files. This is done by putting in place an OpenSocial container which plays a proxy between Graasp's API and third-party applications [2].

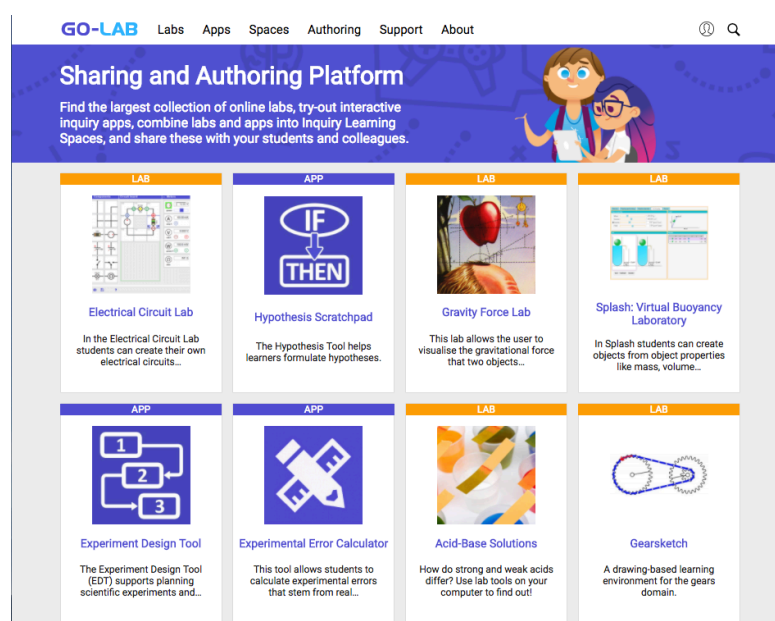


Figure 1.3 – The landing page of the Golabz platform, where teachers can find and share apps, labs and ILS.

⁶<http://www.golabz.eu/>

Chapter 1. Introduction

Inquiry Learning Spaces

The ILS is a predefined pedagogical sequence which guides students through an inquiry learning scenario to learn about a given scientific subject. The sequence is made of 5 main phases: *Orientation*, *Conceptualization*, *Investigation*, *Conclusion*, and *Discussion*. The students go through this sequence of phases where they are provided with adequate content. The ILS lets students learn about science the way scientists do. In the *Orientation* phase students are introduced to a subject, in the *Conceptualization* phase students are asked to make hypotheses which would explain the phenomena they observed, in *Investigation* students are provided with tools to experiment and save their results. Later in *Conclusion* the students can analyze their results and discuss their findings in the *Discussion* phase [55].

Figure 1.4 shows the teacher view of an ILS in Graasp. This is where the teacher can add resources and organize them. In this case, the teacher chose to place a CPL and a graphing tool in the *Investigation* phase. Figures 1.5 and 1.6 show the resulting student view where phases are displayed as tabs.

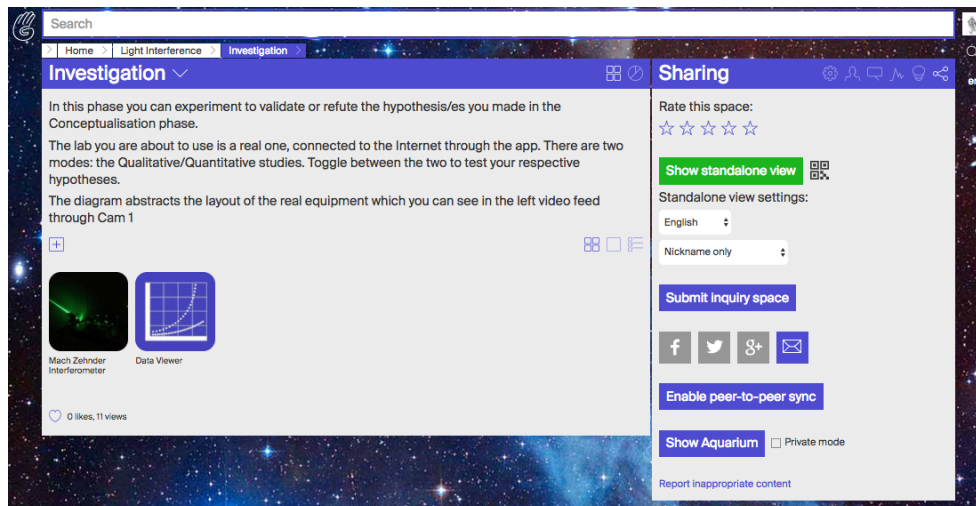


Figure 1.4 – The teacher view of an ILS in Graasp at the Investigation phase, here the teacher used a remote lab and a scaffolding tool

1.2. Educational Web-based Platforms

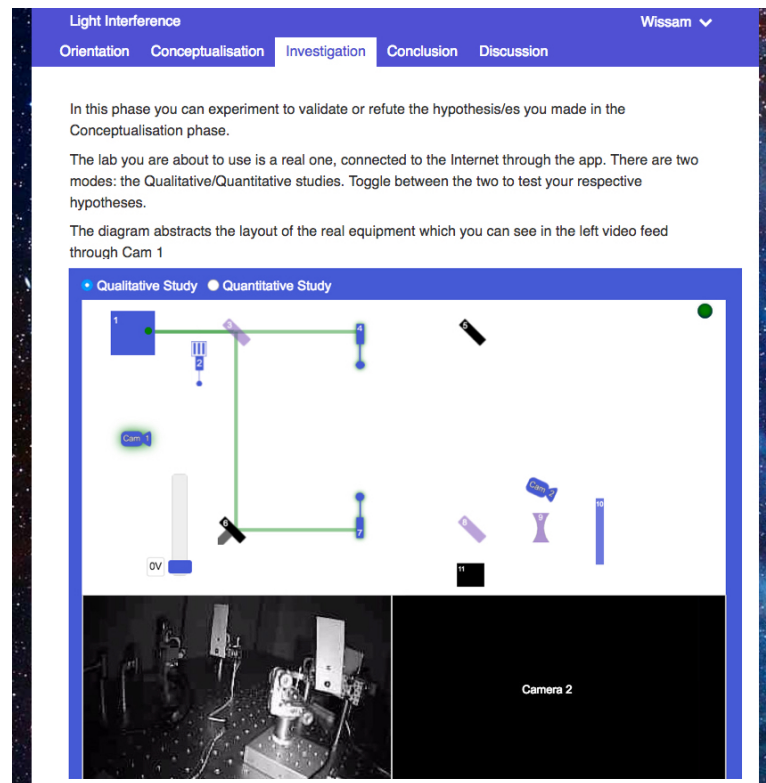


Figure 1.5 – The student view of an ILS at the Investigation phase which shows an embedded remote lab.

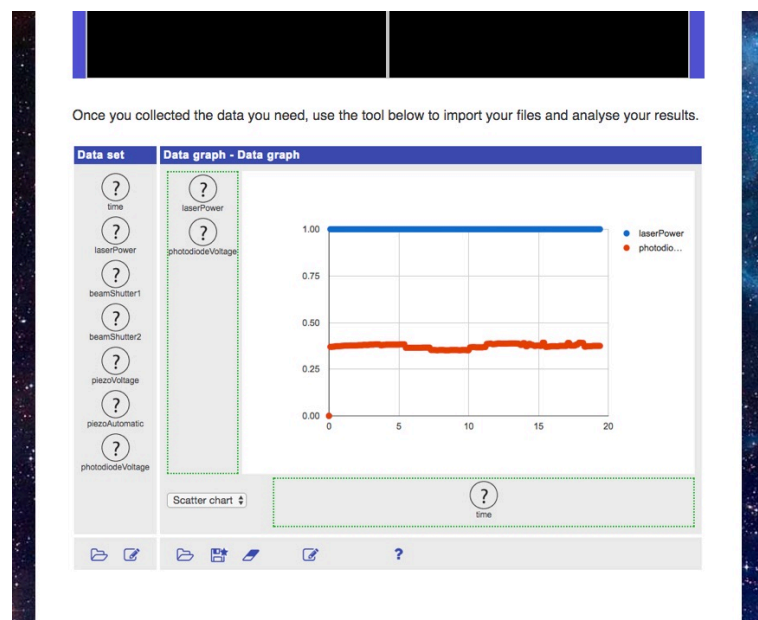


Figure 1.6 – The student view of an ILS at the Investigation phase which shows a scaffolding tool (graphing tool).

1.2.2 MOOC Platforms

A Massive Open Online Course is a course hosted on an online learning environment referred to as a MOOC platform, it is open to anyone to join and large numbers of students are expected to enroll in it. It is claimed that the first MOOC platform appeared in 2007: the ALISON⁷ (Advance Learning Interactive Systems ONLINE) by Mike Feerick [58]. In an interview with Forbes, Feerick says that his main motivation is “spreading education more broadly”. To remain a free platform, ALISON is not affiliated with any university, because it is expensive to be aligned with a traditionally accredited model, and content is submitted by anyone and curated by ALISON [41]. Consequently, the majority of students on ALISON are from the developing world [79]. In 2008, George Siemens, Stephen Downes, and Dave Cormier (who is credited for coining the term MOOC) release the first cMOOC (connectivist MOOC) by the University of Manitoba [41]. cMOOCs are based on the social constructionist paradigm, where the participants in the same class are at the same time teachers and students who learn from each other through interaction, for example in discussion forums and blogs. Then in 2012 the xMOOC platforms appeared: Udacity by Thrun of Stanford, edX as a joint effort from Harvard and MIT, and Coursera. xMOOCs are very much like university courses, they are given by instructors who record short sequences of videos, provide lecture slides and grade quizzes [41].

In this thesis, we consider the case of a Control Systems Lab MOOC⁸ offered by the Automatic Control Lab at EPFL, on the EPFL instance of Open edX. This MOOC is offered as part of a blended learning approach, where the ex-cathedra lectures for the Control System course are given at the university, and the hands-on sessions are offered online. Figure 1.7 shows an overview of the MOOC. The red horizontal box shows a bar with the course main pages, for example **Home** which leads to the homepage of the course, **Course** which is selected and comprises the complete learning material in a structure, the **Discussion** forum where students can go and ask questions to the instructors or peers, and others. The side vertical orange box shows all the Modules of the courses, which are the individual sessions which students need to complete. And the blue horizontal box highlights the sequence of educational content in Module 2 in this case.

⁷<http://www.alison.com>

⁸https://courseware.epfl.ch/courses/course-v1:EPFL+controlsys+2017_T1/info

1.3. Data collection in Digital Educational Settings

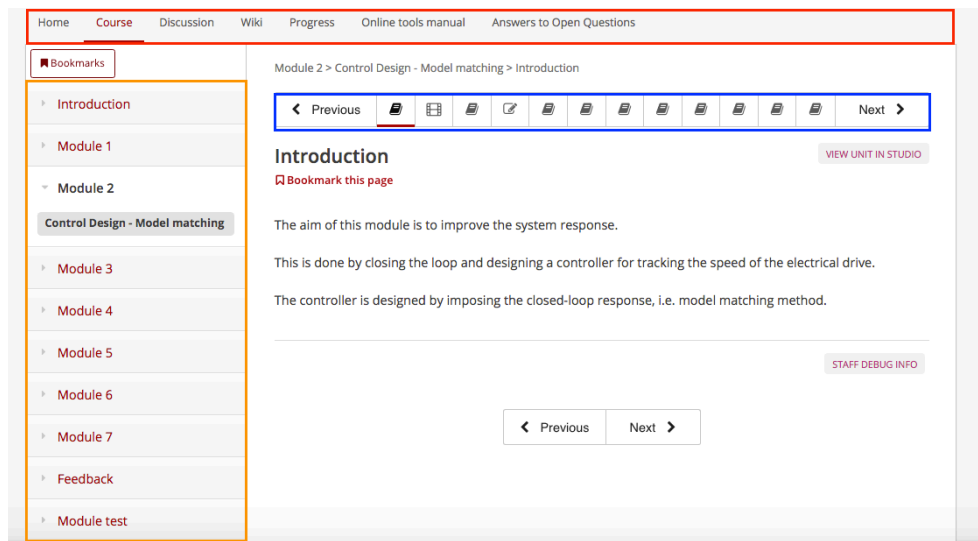


Figure 1.7 – Control Systems MOOC overview. At the top in the red box, you can see the course’s main pages. On the left, you can see the course’s modules. The horizontal blue box show the sequence of the selected module.

1.3 Data collection in Digital Educational Settings

In this section, we discuss issues and current approaches related to the data students produce as a result of their interaction with the CPLs, when embedded in a web platform. First, we talk about the measurements and their use in other tools, second about the activity traces students are leaving behind.

1.3.1 Interaction Continuity

When students are experimenting in a lab room, they almost always keep record of the set of parameters they applied to the lab equipment and the corresponding results. This is either done by hand with pen and paper, or using a computer if the lab’s apparatus is connected to it with a data acquisition cable for example. This data is used for graphing or for archiving, and it is valuable for students to understand what they did and link their results to the theories they learned in ex cathedra classes. When using a cyber-physical lab in an online learning environment, students do the same: they push parameters to the lab through the lab’s web app, and they expect back results which they would want to collect for later use. We discuss two modalities currently used in Graasp and Open edX to support such a mechanism:

In the Go-Lab infrastructure: the learning phases of an ILS shown in Figure 1.5 are the equivalent of folders in the authoring tool Graasp as shown in Figure 1.8. In addition to the five inquiry learning folders for the phases, there are two which are hidden to the students: *About* and *Vault*. The *Vault* folder is destined to be as placeholder in the background of the ILS student view

Chapter 1. Introduction

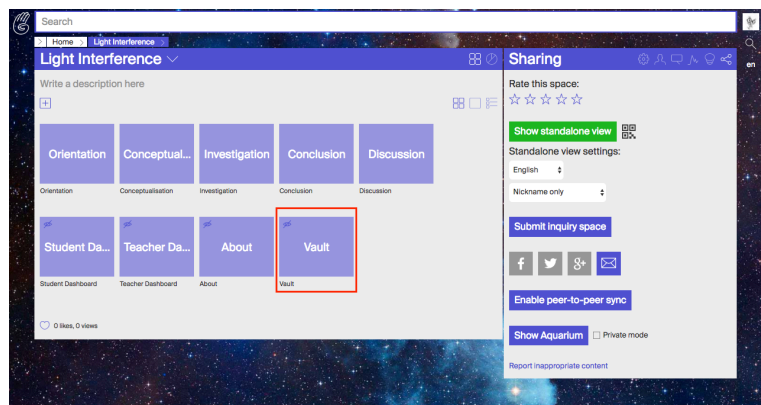


Figure 1.8 – The Vault space destined for saving and retrieving files, only accessible to the teacher

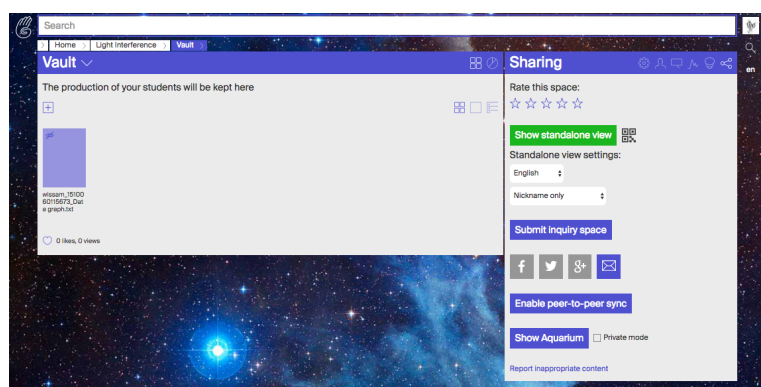


Figure 1.9 – When the Vault is opened, we can see the contents are generated and used by apps in the ILS, in this case it's a single data file.

for data exchange between applications. Figures 1.8 and 1.9 show the *Vault* space in the teacher view and the contents of it respectively. For example, the data viewer app shown in Figure 1.5 uses that file to do the plot.

In Open edX: the platform does not provide any mechanism for saving and retrieving data from and to third-party applications. An ad-hoc solution for supporting the manipulation of student files was put in place: an external database supported the traffic of the measurements from the embedded lab web app to other tools used in the platform. Figure 1.10 shows how the data flows first from the embedded lab web app, then is retrieved in a graphing tool to fit a model to the results, which in turn can be saved to that database.

In these two cases, we say that the continuity of interaction or activity [51] is supported, because the students don't have to leave the online environment to save and retrieve their experimental data in order to use it in other tools.

1.3. Data collection in Digital Educational Settings

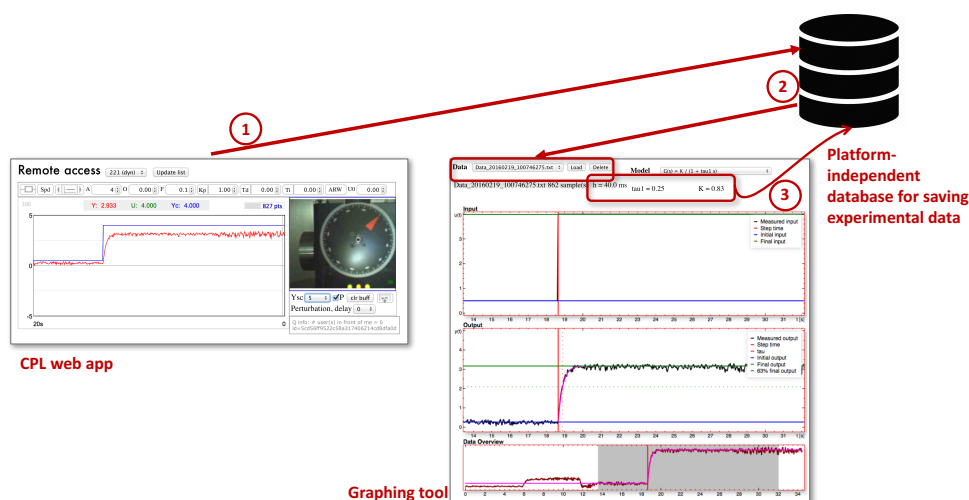


Figure 1.10 – The ad-hoc solution for edX: an external database holds the experimental data and serves to the tools [figure adapted from [62]]

1.3.2 Activity Tracking

Students' interaction with the cyber-physical labs produces large corpora of activity data, which describes their actions on the system through the use of the embedded web app (clicking buttons, moving sliders...). In the context of the Go-Lab infrastructure and in Open edX, students are tracked at different levels of the interaction, either at the level of the platform (which page is opened), or at the level of the learning resource (watching a video), or both. This data is then used to visualize the collective and individual activity of class members. Next we discuss how activity tracking is implemented and utilized in the Golabz infrastructure and Open edX.

In the Go-Lab Infrastructure: in the case of using a CPL embedded in an ILS, a platform-specific solution for collecting the traces of students is used, and is managed by the teachers who decide whether they want to track their students or not. Then the teachers can use the data in custom-made dashboards which show them metrics of students' behavior. While the apps are destined for teacher-use, they can choose to share them with their students. Teachers are responsible for regulating students' privacy, and students only have access to the dashboards if the teacher provides it.

In Open edX: students are tracked with platform-specific mechanisms. Neither the students nor the course instructors can control whether tracking is activated or not, only the platform administrators. The data is available to the instructor, and the platform provides built in dashboards for certain indicators. An example dashboard is shown in Figure 1.12 where instructors can see the breakdown of the number of active learners, watched videos and attempted problems for the week ending on October 15th, 2017. Teachers can also download data from the platform as shown in Figure 1.13, where it is possible to download in a CSV format the profile of students, responses to problems and others.



Figure 1.11 – An example learning dashboard in Graasp made of two activity visualization apps: (1) the Active Users app on the top which shows in real-time which students are in which phase of the ILS; and (2) the Time Spent app in the bottom which shows the time spent per student in the each phase of the ILS [Source: [76]].

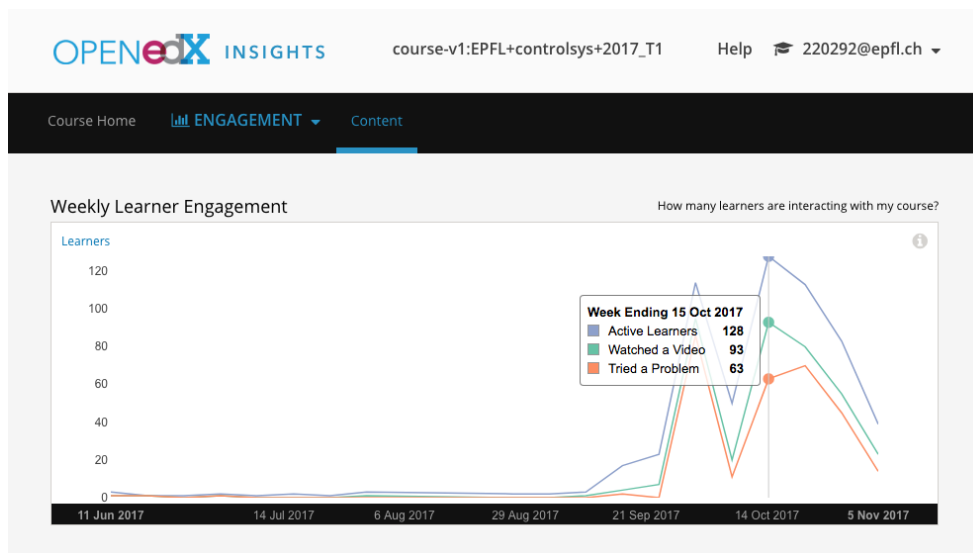


Figure 1.12 – Example dashboard provided by Open edX showing student engagement on the week of October 15th, 2017. It shows the number of active learners, videos watched and problems attempted by students.

Reports

For large courses, generating some reports can take several hours. When report generation is complete, a link that includes the date and time of generation appears in the table below. These reports are generated in the background, meaning it is OK to navigate away from this page while your report is generating.

Please be patient and do not click these buttons multiple times. Clicking these buttons multiple times will significantly slow the generation process.

Click to generate a CSV file of all students enrolled in this course, along with profile information such as email address and username:

[Download profile information as a CSV](#)

Click to generate a CSV file that lists learners who can enroll in the course but have not yet done so.

[Download a CSV of learners who can enroll](#)

To generate a CSV file that lists all student answers to a given problem, enter the location of the problem (from its Staff Debug Info).

Problem location:

[Download a CSV of problem responses](#)

Click to list certificates that are issued for this course:

[View Certificates Issued](#) [Download CSV of Certificates Issued](#)

Click to generate a CSV grade report for all currently enrolled students.

[Generate Grade Report](#) [Generate Problem Grade Report](#) [Generate ORA Data Report](#)

Figure 1.13 – A screenshot of the data download option for instructors in Open edX where they can choose from a variety of data structures and formats.

1.4 Educational Data Analysis

Two main scientific and research communities were born around the same goal of collecting and processing the massively available data through the interaction with educational software, to quantify previously qualitatively evaluated hypotheses in pedagogy, to evaluate learner's progress, to provide students with actionable insights regarding their behavior online while aiding them in adjusting their learning strategies, and many other purposes [23, 67]. These two societies are LAK (Learning Analytics and Knowledge) and EDM (Educational Data Mining).

A big number of studies in these two communities focus on the learning paths students adopt in their progression in a MOOC for example, in order to predict or model successful vs. failing students behavior. In these studies, students are tracked on the platform level where their location in the learning path is detected and sequenced. But the granular activity manifesting in the stages students are jumping between, watching a video, then reading lecture notes, then doing the quiz for example; is not used and integrated. Matter of fact, the learning resources usually used in online learning are rather passive to the students. Students can watch a video but not interact with it, the outcome is just watching it. Whether the students understood or not what's in that video is left to inference, possibly through indicators which the researcher can design and evaluate. Assignments are also passive learning resources if the only thing required from the students is to input the answers. Evaluation methods such as quizzes have the same characteristics.

1.5 CPLs as Modular Open Educational Resources

So far, we have presented what are cyber-physical labs and how they are used when embedded in an online learning platform. The process of developing and deploying cyber-physical labs before they can be utilized as presented, is composed of different stages. How each of these stages is designed and implemented affects how a CPL can be used and reused, which eventually influences the adoption of CPLs in digital education. In this section, we describe our adopted approach for building and integrating cyber-physical labs in online learning environments as modular, portable and context-aware Open Educational Resources (OER).

1.5.1 Background

UNESCO defines Open Educational Resources as being “*any type of educational materials that are in the public domain or introduced with an open license. The nature of these open materials means that anyone can legally and freely copy, use, adapt and re-share them. OERs range from textbooks to curricula, syllabi, lecture notes, assignments, tests, projects, audio, video and animation.*” [74]. Teachers can abundantly find online various electronic OERs such as documents, interactive web applications, videos, cyber-physical labs and others from different sources (Google, online educational repositories such as Golabz⁹, OER Commons¹⁰ and others). Teachers gather and structure relevant resources to carry out a learning activity such as a lesson in a MOOC, or an ILS as it detailed in Section 1.2.1, and then share the curated content with students.

As opposed to how presented so far, in some cases an online lab experiment is conducted separately from pedagogical contexts (lessons), and web-based learning environments are not prepared to fully integrate CPLs. For CPLs to be fully integrated in a platform, we claim that they should be able to: (i) retrieve information regarding the context (where they are embedded), (ii) provide action logging, and (iii) save and retrieve data.

Existing cyber-physical lab solutions are in the form of standalone applications or web applications. A basic solution for integration in learning environments is wrapping the CPL web app in an HTML iFrame. This poses a number of challenges to attain the integration goal. In this section, we define the requirements for the design and implementation of educational cyber-physical labs regardless of the target embedding platform. The proposed formalization of the integration layers for CPLs as Open Educational Labs (OEL) is part of the standardization efforts expanded in the IEEE Networked Smart Learning Objects for Online Laboratories Working Group (NSLOL WG), for the P1876 Standard for Networked Smart Learning Objects for Online Laboratories¹¹.

⁹<http://www.golabz.eu/>

¹⁰<https://www.oercommons.org/>

¹¹https://standards.ieee.org/email/2012_09_cfp_P1876wg_web.html

1.5.2 Formalizing the Integration of CPLs in Educational Web-Based Platforms

Our proposed architecture is based on the concept of separation of concerns, where the system is composed of interconnected yet independent components (the modules or layers), which communicate through defined interfaces. To this end, our architecture is three-layer, is depicted in Figure 1.14 and detailed hereafter.

In the **first layer**, the physical equipment of the CPL is abstracted as a set of software services, based on the Lab as a Service (LaaS) paradigm [70, 73]. LaaS is a term derived from the XaaS series of terms, where “X” means everything and “aaS” refers to “as a Service” [43]. Following this paradigm, the assumption is that everything “X” is offered as a service over the internet rather than at a physical space. It is a notion derived from Service Oriented Computing (SOC), where software is made available as a set of services, and hence hiding the dynamics and only exposing the program through a well-described API (Application Programming Interface) [22]. “LaaS” refers to Laboratory as a Service, where a laboratory is abstracted and made remotely available through the internet as a software service. Building a cyber-physical lab according to the LaaS paradigm should result in well-defined APIs to access the lab’s apparatus from a user application. This enables the independence between the two tiers of the traditional Client-Server architecture adopted for cyber-physical labs, where typically the *Server* side is composed of the lab apparatus and the interfacing software, and the *Client* side is the user application through which interaction with the CPL is possible. This separation of the *Client* and the *Server* enables the personalization of the user web app (the Client), which in our architecture is sitting in Layer 2 of Figure 1.14. The API provides a set of routines to read and write data from and to the cyber-physical lab respectively. The basic implementation should accept requests for data retrieval from the sensors reflecting the state of the lab, and writing data requests on actuators for controlling the lab. The lab as a service is a self-contained layer that is operational regardless of the web app or the hosting platform. Next, in the **second layer**, the cyber-physical lab is ready to be personalized as an Open Educational Lab (OEL). At this layer, the web app should provide the users with an interface to interact with the lab’s apparatus and conduct their experiments. Requests such as actuator control and sensor reading should be implemented by invoking the API calls of the LaaS to gain access to it. At this stage, the cyber-physical lab can be exploited without any context (i.e. without being part of a lesson). But if chosen to be used in a pedagogical scenario, connecting the web app to the LaaS, and augmenting it with user *identity management*, *activity tracking*, and *experimental data management* turns it into an OEL ready to be integrated in a hosting platform (**layer 3**).

1.5.3 Open Educational Labs

Cyber-physical labs are interactive educational resources, where user-action has an effect on the system, and which generates data belonging to two categories: *interaction data* resulting from the use of the web app, and *experimental data* which are the data sent to the actuators and received from the sensors of the CPL. Collecting the generated data and linking it to the originating user is

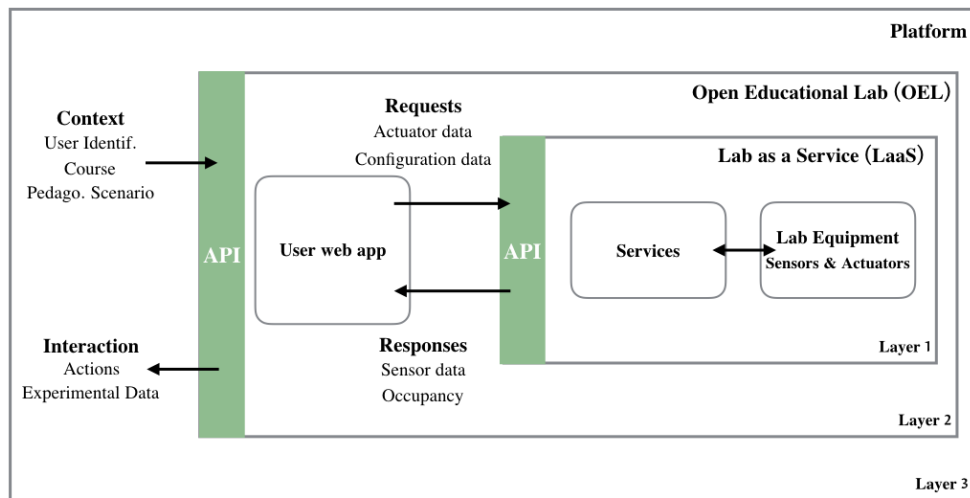


Figure 1.14 – Formalization of the integration layers for cyber-physical labs in online learning platforms as modular OERs.

important for a number of goals: the generated data from the interaction with the web app UI (User Interface) components (buttons and sliders for example) of the web app is valuable for studying interaction patterns for example, and experimental data are needed by the learners to check their results and possibly use them in other tools as discussed in Section 1.3.1. In order to support the full integration of CPLs in target platforms, there is a need to specify the requirements and accordingly develop CPLs as OELs. Hence, we define Open Educational Labs as being Open Educational Resources which are augmented with **access management**, **activity tracking**, and **data storage** functionalities which guarantee a full integration of the CPLs in a hosting platform. Below we detail these three requirements:

Access management: in this thesis, we are interested in the case where a CPL is part of a complete educational activity (i.e. a lesson). We assume that learners connect to the learning activity through an online learning platform, hence having a user identity for authentication with the platform (unless an anonymous mode is used for example). Referring to Figure 1.14, we consider that the CPL will be integrated in the platform through an interfacing module, which in most cases is a third-party application. To prevent the creation of multiple identities belonging to the same user, it is necessary to propagate the user identity from the platform, to the OEL as in Single-Sign On (SSO) for instance. More specifically, when learners are conducting their educational activity, they should have a unique identity that persists throughout the different sessions and the integration layers. This guarantees the consistency of reflecting the contexts, saving activity traces, and collecting experimental data. In our proposal, a user authenticates with the platform to get access to the OEL, the OEL authenticates with the LaaS to get access to lab.

User activity tracking: when a CPL is first abstracted as a LaaS, then as an OEL to be integrated in a learning platform, we see that there are several sources of activity traces. At the platform level, the log in and log out times indicate how much time a learner spent in the lesson for example. At the LaaS level, keeping records of the different exchanged requests and responses with the web app can help in bringing meaningful insight into lab usage. Precisely, the experimental parameters can be used to extract usage patterns for a certain experiment and hence understand how students are using the lab when studying a certain concept.

Data storage and retrieval mechanisms: when conducting an experiment, students generate the results of applying parameters on the process of the given lab which they would use for graphing or tabulating results for example. To improve user experience as discussed in Section 1.3.1, it is necessary to specify mechanisms for data saving and retrieval.

1.5.4 Proof-of-concept Implementation

In this section, we present the example of the Mach-Zehnder interferometer CPL (detailed later in Section 1.6.1), developed and integrated in Graasp as an OEL according to the proposed guidelines.

The LaaS Layer (Layer 1)

Figure 1.15 depicts the first layer of the Mach-Zehnder as an OEL. The lab equipment is interfaced through a myRIO ¹² to allow software access to it. The services which allow the web client to control the lab's apparatus are implemented as APIs on the same board.

The OEL Layer (Layer 2)

In the second layer, the web app provides a web app for students to use the Mach-Zehnder interferometer. The web app has figure-based components for the interaction with the equipment. By clicking on the UI elements of the web app, the users will be altering the state of the lab and experimenting. For example, the photodiode is abstracted as a rectangular box which students can click to turn ON and OFF. The cameras are abstracted as camera-shaped icons, when students click them they are activated, a green halo appears around the corresponding icon and the live feed is transmitted. The web app connects to Layer 1 through the APIs served by the myRIO board.

¹²The myRIO board is a reconfigurable I/O embedded computer board which supports signal acquisition and generation and network access.

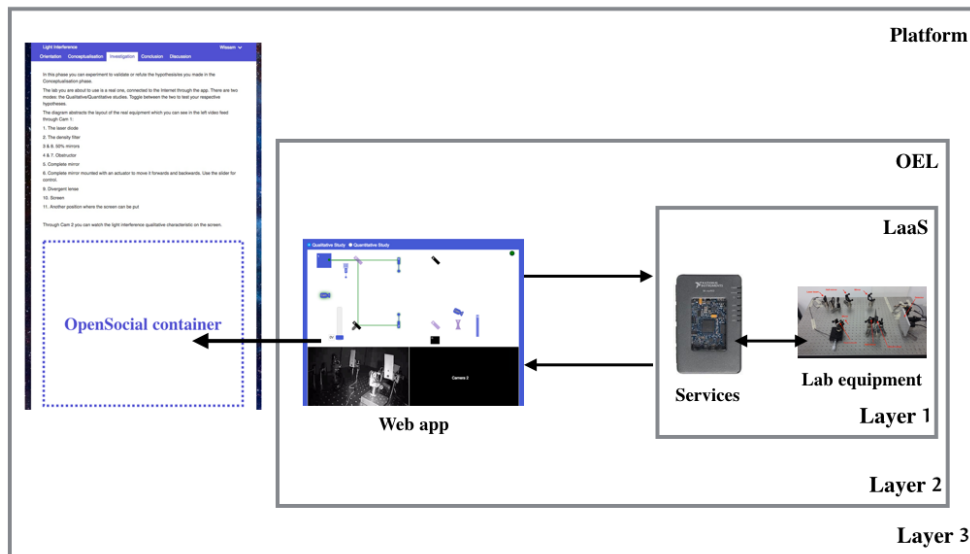


Figure 1.15 – The 3 layers of integration for the Mach-Zehnder interferometer as an OEL.

The Integration in Graasp (Layer 3)

To implement the three requirements for integration: access management, activity tracking and saving and retrieving data, we make use of the OpenSocial container in Graasp (Section 1.2.1). Because this example lab is integrated in an ILS, we also make use of the ILS library¹³ which, using the OpenSocial API takes care of ILS specific mechanisms such as identifying in which phase of the inquiry learning sequence the web app is embedded through a one-liner. In layer 3 of Figure 1.15, the integration of the OEL in Graasp within a learning activity in the ILS is shown. In addition to communicating with the LaaS layer, the web app is aware of the user identity through Graasp's *People API* and saves associated activity tracks through the *ActivityStreams API* and experimental data through the *Documents API*.

¹³<https://github.com/go-lab/ils>

1.6 Running Examples

In this section, we present two CPL examples which are used throughout the thesis for solution prototyping and validation. We start with the Mach-Zehnder Interferometer example, followed by the Electrical Servo Drive.

1.6.1 The Mach-Zehnder Interferometer

The Mach-Zehnder Interferometer (MZI) is a highly configurable device built by and named after the physicists Ludwig Mach and Ludwig Zehnder in 1892. It is used to study a number of fundamental topics in classical physics such as light interference, and in quantum mechanics such as counterfactual definiteness, quantum computation, quantum cryptography, quantum logic and Elitzur-Vaidman bomb tester, in addition to many others [56]. For example, in optical telecommunications it is used as an electro-optic modulator for phase and as an amplitude modulation device for light that transports the information. The configuration of the Mach-Zehnder Interferometer considered in this thesis is to study light interference phenomena by means of division of a collimated light beam.

The Mach-Zehnder interferometer is rarely introduced to students in high schools, since usually quantum physics are not part of the curriculum, consequently the device is not mentioned in textbooks and high school teachers are not familiar with the experiment. Recently, quantum physics was added as an optional topic in Swiss high-schools. For many students the conceptual assimilation of quantum mechanics can be rather hard, owing to the counter-intuitive nature of quantum phenomena [56]. By providing a tool that makes understanding quantum physics an easier task, the intimidation caused by the difficulty to apprehend quantum phenomena is surpassed, and hence contributing to the efforts of encouraging students to pursue scientific and engineering majors. Two teachers at Gymnase de Morges¹⁴ acquired the equipment to build an MZI, and they designed several experiments around the topic of light interference.

The basic MZI is used to demonstrate light interference by division of a light beam. Figure 1.16 depicts the MZI layout. The device is composed of a light source collimated or coherent, two beam splitters (BS1 and BS2), two complete mirrors (M1 and M2) and two detectors or screens (D1 and D2). Beam splitters are optical components which split an incident light beam into 2 or more, with equal or different light intensities [25]. The beam splitters used in this example are half-mirrors which reflect 50% of the incident light and transmit the remaining 50%. The complete mirrors reflect 100% of the incident light. In this device, the interference is a result of the phase difference between the laser components introduced by the combination of mirrors and beam splitters [56, 85], as it will be further detailed later. Looking at Figure 1.16, the following events occur once the laser diode is turned on:

¹⁴<http://www.gymnase-morges.ch/>

Chapter 1. Introduction

1. A laser beam first hits the beam splitter BS1. BS1 causes the light to split into 2 light beams: the transmitted beam traveling from BS1 to M1, and the reflected beam traveling from BS1 to M2.
2. The incident beam at mirror M1 is 100% reflected and takes the path between M1 and BS2. The same goes for the reflected beam which hits mirror M2, fully reflects and takes path M2–BS2.
3. Both beams reflected by M1 and M2 hit the beam splitter BS2 and split to 2 paths, one transmitted and the other reflected. The resulting beams hit detectors D1 and D2.

The Mach-Zehnder Interferometer is an example of an experiment which is hard to acquire by schools due to the expensive material needed for it, and not all teachers have the knowledge to explain such phenomena and teach them. Hence our collaboration with the people in Gymnase de Morges to help spread the accessibility to such knowledge and lab setups.

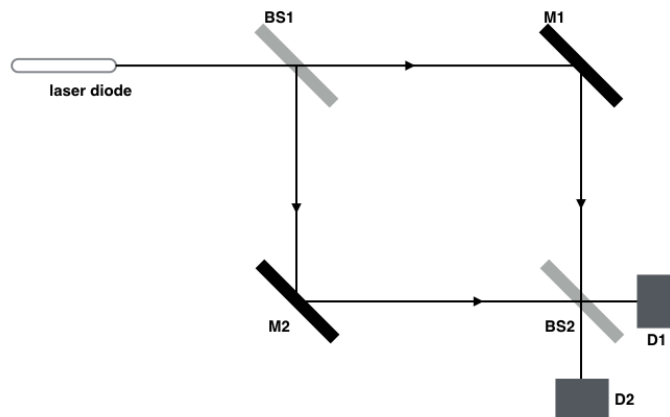


Figure 1.16 – Basic Mach-Zehnder interferometer layout

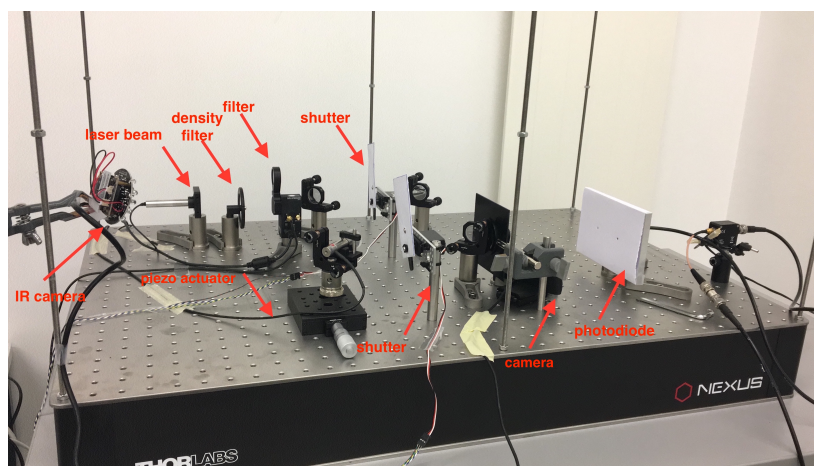


Figure 1.17 – Picture of the Mach-Zehnder interferometer.

1.6.2 The Electrical Servo Drive

The electrical drive depicted in Figure 1.18 is used for the experiments of the Control Systems Lab MOOC. The system is used to conduct a series of experiments through the course to both learn about automatic control and understand the limitations and constraints of real physical systems. The students are meant to master the implementation of a PID controller by experimenting with this servo system remotely, after their design of the controller, and precedes the validation step of the controller design. Of the control principles to learn throughout this course: linear ranges of systems, how to do system modeling in the time and frequency domains and others.

The physical system is composed of two motors lodged inside the box mounted on the red cylinder as base, a visualization black disc which is an indicator of the position and a metal load on the side which is driven by the motors. There is a camera which transmits a live feed of the disc and there are sensors which are better seen in Figures 1.19 and 1.20.

Looking at Figure 1.20, we see one of the two motors acting as a signal generator and controlling the second, which is connected through gears to the load. There are 2 sensors: one which collects the value of y_ω (rotation speed of the load), and another which captures y_θ (the rotation angle of the load). The students have a total of 11 parameters they can control.

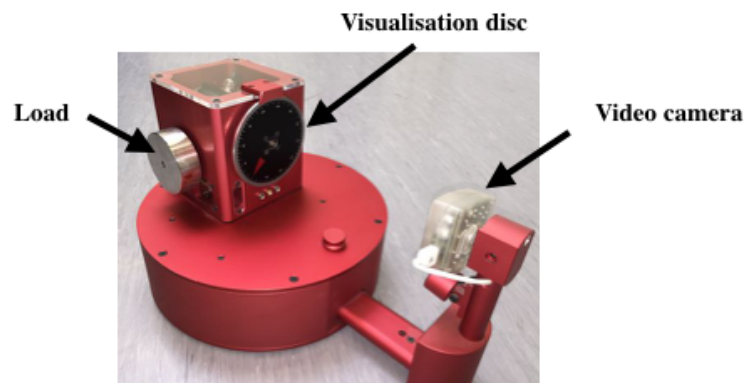


Figure 1.18 – The electrical servo drive system used by the students of the Control Systems Lab course [annotated from [64]].

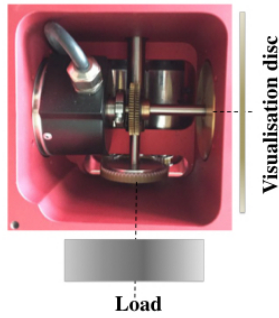


Figure 1.19 – Top view of the box where the electric servo drive is lodged [annotated from [64]].

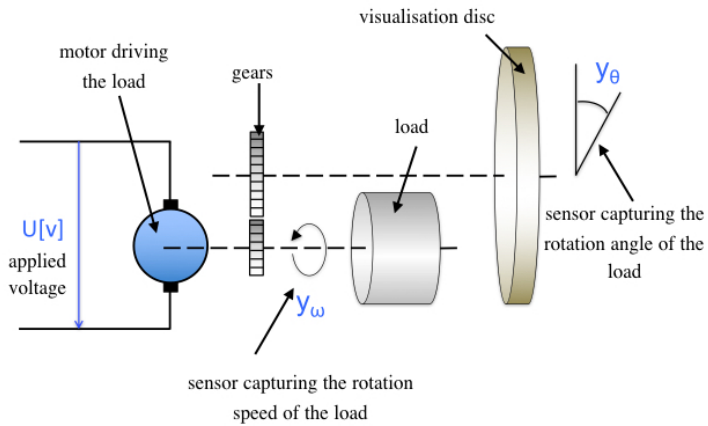


Figure 1.20 – A graphical representation of Figure 1.19 [annotated from [64]].

1.7 Challenges, Contributions and Validation

There is a number of stakeholders for CPLs embedded in an online learning environment: **students** as users, **teachers** as content curators, and **lab owners** as service providers. Each stakeholder faces challenges to either have or provide a better experience using CPLs. In this section, where appropriate, we formulate the user stories (US) of these stakeholders to provide an overview of their needs. User stories are short informal descriptions of desired features in a system from the perspective of the end user. User stories usually follow the user-story template [68]:

“As a <role>, I want <goal/desire> so that <benefit>”

Then we elaborate our problem statements pertaining to each of the user stories, formulate them as research questions (RQ) and briefly detail the associated contributions.

1.7.1 Enabling the Automatic Generation of Web-Apps for CPLs

When **teachers** are preparing a lab session, they go to the lab room and configure the equipment to conduct certain experiments in relation to what they explained in ex-cathedra lectures. Sometimes, teachers use the same lab equipment or a subset of the components in different configurations to conduct different experiments. Configurable cyber-physical labs are a type of labs with which many experiments can be done using the same apparatus, by permuting through possible configurations of lab components. When using cyber-physical labs in their online lessons, teachers expect to be able to do the same (configuring the lab for different experiments), hence our first user story:

US1: As a **teacher**, I want to be able to configure my cyber-physical lab experiment the same way I do with physical labs, regardless of my absent or limited coding skills.

Which leads to the first research question:

RQ1: How can we support teachers in the generation of web apps for configurable cyber-physical labs, without the mediation of an application developer and the lab provider?

The goal is to support **teachers** in choosing the experiments and creating the respective web app on their own, in a pedagogically oriented scenario and by taking into consideration the target online learning environment. We revisit the Smart Device Specification¹⁵ and extend it by describing the possible “*configurations*” or “*experiments*” of labs supporting one or various experiments (**contribution 1a**), further enabling the automatic generation of the interfacing web apps. To that end, we add a service description to the mentioned specification, which returns the configurations or experiments supported by the cyber-physical lab, in addition to the requests and responses data models.

To verify the feasibility and completeness of the extension, we propose a tool for generating the web apps (**contribution 1b**) which are ready to be integrated in two educational platforms: Graasp and any platform which supports LTI¹⁶ (Learning Tools Interoperability specification). Furthermore, we evaluate the automatic web app generator with the Mach-Zehnder Interferometer.

1.7.2 Activity Tracking Infrastructure for CPLs

Lab owners as service providers are interested in understanding how their labs are utilized and used by the users (students), hence their interest in tracking the activity occurring on their systems. On the other hand, **students** as individual users of the lab are interested in keeping record of their experimental results, and as members of a class are interested in being aware of the progress of others, hence the need to adequately collect learning data and presenting them to students in a valuable fashion as illustrated in the following user stories and corresponding research questions:

US2: As a **lab owner**, I want to be able to track users of my systems, so I can understand how they are used and utilized, provide better services, and advertise them online.

US3: As a **student** I want to be able to save my experimental results so that I study or archive them for my personal use.

US4: As a **student** participating in an online class, I want to be able to know the status of others using the CPLs in the same online class, so I can be aware of what is happening and be able to reflect and adjust my strategies in case I am stuck.

US5: As a **student** using a cyber-physical lab, I want to be able to control my data privacy so I

¹⁵The Smart Device Specification describes how a CPL should be implemented to cancel the dependencies between the user web app and the lab server of the system, and provides guidelines for CPL API design. More details in Chapter 2.

¹⁶<https://www.imsglobal.org/activity/learning-tools-interoperability>

am sure my data is not shared with undesired audience or used inappropriately.

The 3 research question which follow are:

RQ2: What are the experience data that students and lab owners are interested in?

RQ3: What are students' online privacy concerns?

RQ4: How to design an activity tracking infrastructure which responds to the needs of both students and lab owners, while preserving students' privacy?

Existing solutions for activity tracking in educational web environments capture the interaction with the context (e.g. space in Graasp, and tab in edX lesson) rather than with the embedded resources, in our case a CPL. In this thesis, we focus on the needs of *students* and *lab owners* as stakeholders of the collection, storage, and retrieval of activity traces, taking into consideration the differing needs.

Based on an elicitation and analysis of system requirements, we propose an infrastructure to collect, save, and enable the retrieval of an activity (traces) and the results of that activity (experimental data) in a platform-independent architecture (**contribution 2**). Furthermore, we propose a vocabulary for formalizing the activity traces, the vocabulary makes sure that the traces are well-described in order to enable the extraction of relevant insight for both students and lab owners (**contribution 3**). The infrastructure enables students to regulate their data privacy: where it is saved and who can use it for which purposes (**contribution 4**), the proposed privacy regulation modality relies on both the clear privacy needs defined by the students to comfortably employ the activity learning infrastructure and current official regulations for data privacy.

To validate the feasibility of implementing and using the proposed architecture for both students and lab owners, we respectively build it on the web app side for the Control Systems Lab course and on the lab server side for the Mach-Zehnder interferometer.

1.7.3 Understanding How Students Access and Use CPLs

The interaction data generated from the use of cyber-physical labs enables us to study various aspects of experimentation in the educational context. From one side, in MOOCs where large number of students are expected to be taking a course with a CPL for example, the allocation of limited resources (CPL setups) for massive concurrent access is a challenge for the lab provider. From another side, how students access the labs and use them can give educators an insight on their experimental behavior. We consider the case of the Control System Lab MOOC, where a farm of 25 CPL setups were arranged to service an expected number of 200 students. The collected interaction data can help the lab owner in evaluating the implemented resource allocation strategy and actual need for it, and the course instructors to depict indicators for academic performance as described in these two research questions:

RQ5: How are students accessing and using cyber-physical labs made available to them 24/7?

RQ6: How does students' experimental behavior impact their academic performance?

We collect the interaction data from the use of the CPL embedded in the mentioned MOOC, and we first analyze the access trends of the students to the CPLs (**contribution 5**). Then we mine their experimental behaviors to unveil group-specific patterns. And last we try to find statistically significant indicators from both the access and the use analysis, of what is affecting their academic performance (**contribution 6**).

1.8 Thesis Outline

The rest of the thesis is organized as follows:

- In **Chapter 2**, we detail the contributions pertaining to **RQ1** by extending the Smart Device Specification and implementing an automatic web app generator for cyber-physical labs.
- In **Chapter 3**, we elicit the needs of students and lab owners as stakeholders for an activity tracking infrastructure for cyber-physical labs. We propose an architecture and a vocabulary for collecting, saving and retrieving learning traces with a privacy regulation mechanism. In this chapter we answer **RQ2**, **RQ3** and **RQ4**.
- In **Chapter 4**, we study the behavior of students in a MOOC to reveal the effect of their online learning behavior on their academic performance (**RQ5** and **RQ6**).
- In **Chapter 5**, we conclude and elaborate on possible future works.

Funding Acknowledgments

This thesis was partially funded by the European Union in the context of the Go-Lab project (grant agreement no. 317601) under the ICT theme of the 7th Framework Programme for R&D (FP7), the Next-Lab project (grant agreement no. 731685) Horizon 2020 Research and Innovation Program, and the Swiss National Science Foundation in the framework of its SCOPES (Scientific co-operation between Eastern Europe and Switzerland) program (grant agreement no. IZ74Z0-160454).

2 The Smart Device Configurations

Lab setups which can be re-purposed (similar to the Mach-Zehnder Interferometer presented in Section 1.6.1) are used to conduct many experiments by reconfiguring their components. When using real physical lab setups, teachers switch between experiments by doing the reconfiguration or instructing the students on how to do it. Emulating this functionality when using a cyber-physical lab is a challenge due to several factors involving the system's architecture.

Various architectures enabling CPL systems exist, the most common follow the Client-Server topology. In this context, the Server interfaces the physical setup and makes it software-accessible, and the Client provides a software application for the users to command the CPL through a connection to the Server. The adopted architecture of the respective systems dictates how the Server and the Client communicate, and the extent of the dependencies between these components. At one end of the spectrum, the Server and the Client are built-for-each other, or in other words changing one part of the Server entails a necessary change in the Client for the whole system to continue on functioning. At the other end of the spectrum, the system's architecture is modular and allows for several implementations of the Client to operate the lab. When the dependencies between the Server and the Client are canceled, any person with programming skills can make a personalized Client application to access the lab as desired. Or, those programmers can make applications to automatically generate the Client without re-writing the code for each implementation.

The conception of such automatic client application generators is theoretically possible. But in practice, the existing frameworks for implementing the Server side of the architecture do not support such a goal. Even though they provide a description of the available individual equipment components making up a lab and how they can be accessed, and hence allow the selection and grouping of desired system components for an experiment, it is not clear how they are connected (relationships). Currently, teachers rely on the mediated contact with a lab provider to have information about which experiment(s) the considered lab implements, according to possible combinations of the lab equipment components (sensors and actuators).

We define an ‘experiment’ as being the activity during which students can manipulate a set of parameters, and we refer to the combination of the lab equipment components used in an experiment as ‘configurations’. In this sense, a list of sensors and actuators is not enough to make a guided selection of components to create the web app to an experiment. The goal in this chapter is to support teachers in choosing the experiments and generating the respective interfacing web applications on their own, and by taking into consideration the target online learning environment, through answering the first research question:

RQ1: How can we support teachers in the generation of web applications for configurable cyber-physical labs, without the mediation of an application developer and the lab provider?

This chapter is structured as follows: we start by reviewing related work, then we detail the Smart Device Specification upon which we build our proposal. Next, we present our proposed extension to the specification, a proof-of-concept implementation and a tool for generating the CPL web apps. Last, we discuss our findings and conclude.

2.1 Related Work

As some web technologies emerged and died, many architectures for cyber-physical lab systems have been proposed [39, 61]. The most adopted architectures for CPLs are Client-Server based, and with the appearance of the ‘separation of concerns’ paradigm enabling Service Oriented Architectures (SOA), lab providers started building their laboratories following a more modular approach [61, 70]. With such architectures, the access to the CPL setup is done through web services or APIs, where the laboratory server is exposed as a set of services that can be individually invoked through defined interfaces [39, 63, 86]. The main aim of adopting a Service Oriented Architecture for CPLs is to separate the tiers of the cyber-physical lab system, in order to support *personalization* of user applications and *portability* across embedding web platforms.

In [59], the authors make their debut in defining Smart Devices (SDs) motivated by the need to move away from adopting proprietary technologies for building CPLs, and the need to converge towards common conventions for designing and building these systems. Accordingly, they re-engineer the server side by implementing separate services for the different hardware access which are possible for their example lab. In parallel, instead of creating a complete web application or widget, they provide four separate ones for each of the accessible services: a graphing tool, a video feed, a control panel for the system’s parameters, and a tool for saving the experimental data. The users of a CPL can choose any subset or all the provided widgets to use the lab in a ‘metawidget’. An example of what can be done with systems designed this way is depicted in Figure 2.1, where the user interface is disaggregated into mini-apps, and later re-assembled according to a selection of lab components. While this solution is accessible to teachers, any combination between the lab components is possible, possibly without any experimental meaning.

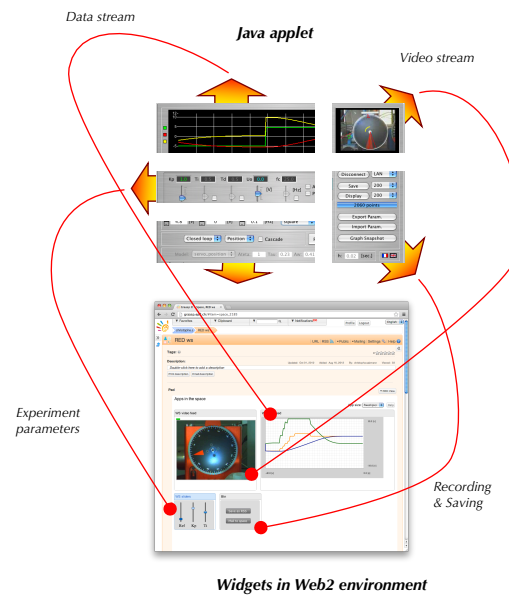


Figure 2.1 – The lab’s web app is disaggregated into mini-apps which can be re-assembled according to a selection of to the lab components they interface. [Source: [59]]

Other frameworks for the generation of cyber-physical lab user web apps exist, such as the tool based on EjsS (Easy Javascript Simulations) [6, 7, 18]. In these works, the authors bring to importance the need for user apps to be well integrated in web-based learning environments such as Moodle. To make this happen, they adapt the tool which was meant to be used with simulations, to be used with real physical lab setups. They introduce a middleware layer between the physical process and the interactive user application generator EjsS, called the JIL server. The JIL server plays the role of a translator between the EjsS tool and the server side of a CPL system, in order to allow the assignment of graphical user interface components to variable or parameters of the physical process. Additionally, they invoke the necessity to support open web technologies and move away from Java applets which are no longer supported by modern web browsers, hence the release of an equivalent Javascript version. While they provide a solution that is reusable, and prevents application developers from building web apps from scratch for each lab, this framework only supports the generation of user clients for labs which are compatible with their implementation of the presented app builder.

2.2 The Smart Device Paradigm and Specification

The Smart Device Paradigm and accompanying Specification are one implementation of the Lab as a Service concept discussed in Section 1.5.2. In their works [59, 61, 63], the authors revisit the Client-Server architecture to cancel the dependencies which inherently exist between these two components. According to Salzmann et al., implementing a cyber-physical lab as a Smart Device enables the personalization of the client web app, and enables teachers to use the CPLs in different ways according to their educational needs, by designing their own pedagogical scenarios as shown in the previous section and illustrated in Figure 2.1.

The Smart Device Specification provides two types of guidelines: the first intended for the communication specifics with the interfacing web app and are required (the *metadata*), and the second for internal mechanisms of the CPL, and are recommended practices (referred to as *functionalities*).

The components of the *metadata* are detailed hereafter:

1. *General metadata*: provides high-level information regarding the CPL such as the name, description and contact information.
2. *API Metadata*: describes the available services (i.e. sensor and actuator services). It is composed of two main sections: *apis* and *models*. The *apis* provides a description of implemented services and how they can be called. The *models* section details the scheme of the requests, responses, and data to be passed to the actuators or sent by the sensors.
3. *Authorization mechanisms*: details the authorization schema to implement at the client side in order to get access to the services.
4. *Concurrent access mechanisms*: provides information on how the lab server manages multiple access requests simultaneously.

And the *functionalities*:

1. *Safe and known state* by performing automatic initialization at startup, resetting the lab to a default state after a client disconnects, etc...
2. *Security and local control* which ensures that malicious accesses do not damage the lab server or the physical equipment. This mechanism includes performing value validation before applying it to an actuator.
3. *Logging and alarms* which will help the lab provider/maintainer to keep track of the use of the lab. For example, in the case of malicious accesses, s/he can identify new request patterns to take into consideration in “Security and local control”.

Based on a comparison between standardized communication protocols, data exchange formats and web services description languages, the choice fell upon WebSocket, JSON and Swagger ¹ respectively, for the formalization of the specification. WebSockets were chosen for their reliance on a duplex communication channel (two-way) ensuring a real-time interaction experience for the user with the CPL. JSON for its elegant format both parseable and readable by machines and humans respectively. Swagger provides a description for HTTP-based web services with a JSON schema and strongly focuses on supporting the automatic generation of user interfaces, hence making it the choice for the Smart Device Specification. In [63] Swagger was extended to describe Websocket-based services.

2.3 The Smart Device Configurations

As mentioned in the previous section, the *API Metadata* of the Smart Device Specification specifies the communication protocol and formats for sending requests and receiving responses from a CPL. In more detail, it is composed of two main sections: *apis* and *models*. The *apis* describe which services are implemented and how they can be accessed, by providing information on the adopted communication protocol, the type of requests to write and responses to receive specified by their corresponding *models*, the parameters to pass to the request, and the authorization schema to implement at the client side if applicable. The *models* section details the structure of the requests, responses, and data to be applied to the actuators or sensed by the sensors. It includes information on the unit, type, allowed ranges, range steps, last measured values, and the value update frequency.

The *apis* section is based on four main API calls: *getSensorMetadata*, *getSensorData*, *getActuatorMetadata*, and *setActuatorData*. *getSensorMetadata* is formatted as a *SensorMetadataRequest* model, returns a list of all sensors in the lab in a response formatted as a *SensorMetadataResponse* model. In the response to this request, the *sensorIds* are included to allow for separate calls to each. To read the data on a specific sensor, the CPL web app calls the *getSensorData* request as modeled by a *SensorDataRequest*, by including the corresponding *sensorId*, as a response the data captured by the sensor is returned in a *SensorDataResponse*. A *getActuatorMetadata* request sent as an *ActuatorMetadataRequest* returns a list of all actuators in the lab: the *actuatorIds* in an *ActuatorMetadataResponse*. The *actuatorIds* is an array which contains the *actuatorId* of separate actuators. To write data to an actuator, it is sufficient to invoke the *sendActuatorData* request formatted as a *SetActuatorData* request, providing an *actuatorId*.

¹<https://swagger.io/swagger-ui/>

Chapter 2. The Smart Device Configurations

The extension to the Smart Device Specification we propose to describe the possible ‘configurations or ‘experiments of labs supporting one or various experiments is two-fold:

1. We define the *models* for an *Configuration*, *SendConfigurationsRequest* and *ConfigurationsMetadataResponse*.
2. Define a new api call: *getConfigurations*.

Configuration model: a *Configuration* model is characterized by 2 fields common to all models: *id* and *properties*. The *id* characterizes the model at hand, in this case its value is *Configuration*. This *id* field gives knowledge to the automatic generator about the format of an *Configuration* JSON object for further processing. The *properties* are made up of 5 sub-fields:

- *configurationId*: which can take any string value. The value of this field is defined by the lab provider.
- *fullName*: which contains a non-formal name of the configuration. It can take any string value.
- *description*: a human readable description of what the configuration is about. This field is meant to be informative for teachers, to get a high level description of the configuration which corresponds to an experiment.
- *sensors*: it is an array containing a list of the sensor ids used in a particular configuration. *sensorIds* can have any string value. The string values of *sensorIds* contained in this JSON object should be corresponding *sensorIds* defined in the *metadata*.
- *actuators*: it is an array containing a list of the actuator ids used in a particular configuration. *actuatorIds* can have any string value. The string values of *actuatorIds* contained in this JSON object should be corresponding *actuatorIds* defined in the *metadata*.

A complete *Configuration* model is shown hereafter:

```

1      "Configuration": {
2          "id": "Configuration",
3          "properties": {
4              "configurationId": {
5                  "type": "string"},
6              "fullName": {
7                  "type": "string"},
8              "description": {
9                  "type": "string"},
10             "sensors": {
11                 "type": "array",
12                 "items": {
13                     "id": "Sensor",
14                     "properties": {
15                         "sensorId": {
16                             "type": "string"}}}
17             },
18             "actuators": {
19                 "type": "array",
20                 "items": {
21                     "id": "Actuator",
22                     "properties": {
23                         "actuatorId": {
24                             "type": "string"}}}
25                 }
26             }
27         }

```

Listing 2.1 – The *Configuration* model as formatted in the extended Smart Device Specification

ConfigurationRequest model: to retrieve the required *actuatorIds* and *sensorIds* for a particular experiment, a *ConfigurationRequest* has to be sent to the Smart Device hosting the laboratory as shown hereafter. The *ConfigurationRequest* should contain the *configurationId* of the desired experiment. A list of *configurationIds* can be retrieved with the *getConfigurations* call.

```

1      "ConfigurationRequest": {
2          "id": "ConfigurationRequest",
3          "required": ["method", "configurationId"],
4          "properties": {
5              "method": {
6                  "type": "string",
7                  "description": "The method should be equal to the nickname of one of
8                      the provided services."},
9              "configurationId": {
10                 "type": "string"}

```

Listing 2.2 – The *ConfigurationRequest* model as formatted in the extended Smart Device Specification

ConfigurationMetadataResponse model: the response of an *ConfigurationRequest* is an *ConfigurationMetadataResponse*. The *id* of this response tells the type of JSON object to expect at the receiving end. It is formatted as to contain the *Configuration* JSON object which defines a configuration. This should be enough for an automatic generator to make a web app corresponding to the required request.

```
1      "ConfigurationMetadataResponse": {
2          "id": "ConfigurationMetadataResponse",
3          "properties": {
4              "method": {
5                  "type": "string"},
6              "experiments": {
7                  "type": "array",
8                  "items": {
9                      "$ref": "Configuration"
10             }}}}
```

Listing 2.3 – The ConfigurationMetadataResponse model as formatted in the extended Smart Device Specification

getConfigurations api: The *getConfigurations* api allows the retrieval of a list of supported configurations. The *nickname* of this call is “getConfigurations” which means it needs to be used when initiating a request. *summary* and *notes* fields give a high level description of what this call does: answers with a JSON object containing the list of available configurations ids. The response of this call is formatted as an *ConfigurationMetadataResponse* which will be detailed later in this section. As it can be deducted from the *properties* field, the request is formatted as a *SimpleRequest* defined in the original Smart Device Specification. The *authorization* field designates authentication mechanisms that the cyber-physical lab is using to permit users to access the lab, if empty it means no authentication needs to be done. *responseMessages* detail the possible responses that can be received at the requester end, in case an *ConfigurationMetadataResponse* cannot be received.

2.4. Validating and Evaluating the Smart Device Configurations with the Automatic Web App Generator

```
1      "method": "Send",
2      "nickname": "getConfigurations",
3      "summary": "Returns a list of possible experiments",
4      "notes": "Returns a JSON array with all the ids of possible experiments",
5      "type": "ConfigurationMetadataResponse",
6      "parameters": [{
7          "name": "message",
8          "description": "The payload for the getConfigurations service.",
9          "required": true,
10         "paramType": "message",
11         "type": "SimpleRequest",
12         "allowMultiple": false
13     }],
14     "authorizations": {},
15     "responseMessages": [{
16         "code": 402,
17         "message": "Too many users"},{
18         "code": 404,
19         "message": "Experiments not found"},{
20         "code": 405,
21         "message": "Method not allowed. The requested method is not allowed
22         by this server."},{
23         "code": 422,
24         "message": "The request body is unprocessable"
25     ]}]
```

Listing 2.4 – The getConfigurations api as formatted in the extended Smart Device Specification

2.4 Validating and Evaluating the Smart Device Configurations with the Automatic Web App Generator

Our claim of the proposed extension for the Smart Device Specification is that it allows the automatic generation of CPL web apps, by teachers, without the need to get help from other entities. In this section, we propose an implementation of an automatic web app generator, to show and evaluate how the formalized extension facilitates our claims.

The automatic web app generator is a tool that enables the production of a fully functional CPL web client in a few clicks. The teacher needs to know the IP address and the port number over which a Smart Device is serving the desired CPL. Using this information, the tool initiates a WebSocket connection with the lab server and subsequently calls the *getConfigurations* service, which returns an array describing each experimental configuration supported by the Smart Device. As mentioned in Section 2.3, each experiment is described by: the *configurationId* that uniquely identifies each experiment, the *fullname* and *description* of the experiment, in addition to the sensors and actuators arrays that contain the ids of all the respective sensors and actuators used by each experimental configuration. These configurations are displayed as checkboxes having the full name and the description of the experiment as their labels. The teachers can then select one or more of the presented possible configurations according to their educational goals. After performing this selection, the automatic generator knows the ids of all the different

Chapter 2. The Smart Device Configurations

sensors and actuators required for each experiment, and will thus send *getAcuatorMetadata* and *getSensorMetadata* requests to the lab server in order to acquire the necessary information about each (see Figure 2.2).

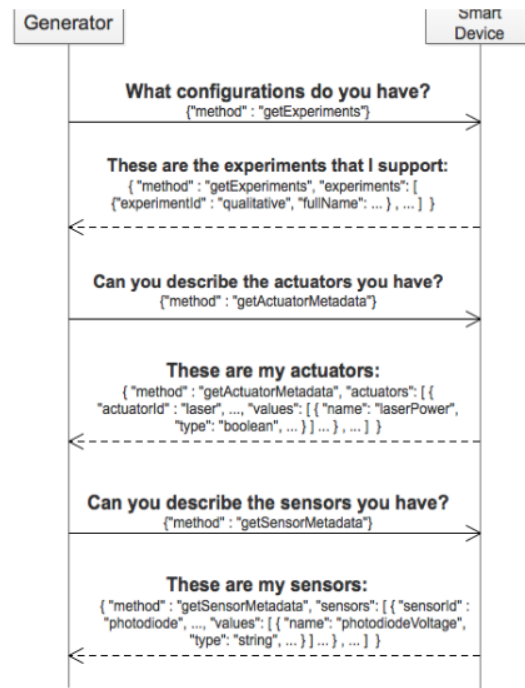


Figure 2.2 – Requests and data exchange diagram between a CPL implementing the extended Smart Device Specification and the automatic web app generator.

For actuator access, the automatic web app generator makes use of some of the fields obtained from the actuator metadata, in order to generate the necessary UI components. It uses the *actuatorId* which uniquely identifies each actuator, to populate the *actuatorId* field of the request packet which is sent to the Smart Device whenever a user of the generated lab client alters the state of an actuator, thus making a call to the *sendActuatorData* service. The automatic generator also uses the *values* field of the metadata, which is an array of all the measurement values each actuator contains. Each actuator value is represented as a separate UI component in the generated widget. The automatic generator uses the following fields from the metadata of each value:

- **name:** used to differentiate among the multiple values of an actuator.
- **type:** used to decide what type of UI component needs to be created for each value. For instance, a value of type 'boolean' will be represented as a button that can be turned on or off by the user. Moreover, a value of type 'float' will be represented as a numeric slider, see Table 2.1.
- **rangeMinimum** and **rangeMaximum:** used by the automatic generator to specify the boundaries of the numeric slider that is created for a value of type float.

2.4. Validating and Evaluating the Smart Device Configurations with the Automatic Web App Generator

For sensor requests, the web app generator uses the *sensorId*, which uniquely identifies each sensor, to populate the *sensorId* field of the request packet that is sent to the Smart Device whenever the lab client makes a call to the *getSensorData* service. The generator also takes into consideration the *webSocketType* field of the sensor metadata to check whether a given sensor requires a text or a binary WebSocket. In case of a binary WebSocket, the generator assumes that it is a video feed and creates a UI component that displays the video. In the case of a text WebSocket, the generator uses the *values* field of the sensor metadata and represents each value as a separate UI component in the generated gadget. The automatic generator uses the following fields from the metadata of each value:

- name: used to differentiate among the multiple values of a sensor.
- type: used to decide what type of UI component should be created for each sensor value. For instance, a value of type 'boolean' will be represented as a LED indicator. Moreover, a value of type 'string' will be represented as a text value.
- unit: used by the generator to append a unit symbol to the retrieved sensor value.

IP Address: 172.22.11.2

Port Number: 8888

Get Configurations

Choose at least one configuration:

- ☒ Qualitative Study : Observing light interference on the screen
- ☒ Quantitative Study : Studying the signal provided by the photodiode

Choose your educational platform:

☒ Graasp ☐ LTI Consumer

Download Gadget Add Gadget to Graasp

You can add the gadget as an app using the following link as App URL:
<http://graasp.eu/resources/582d679b4fcc0ac468d548fd/raw>

Show Instructions

Figure 2.3 – The landing page of the automatic web app generator showing the two available experiment configurations for the Mach-Zehnder lab

Chapter 2. The Smart Device Configurations

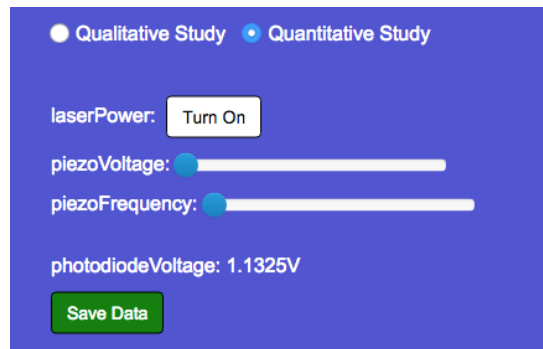


Figure 2.4 – The generated cyber-physical lab client application on Graasp for the Mach-Zehnder lab

<i>Parameter type</i>	<i>Corresponding UI elements</i>
experiment selector	radio button
boolean control	button
range input	slider
sensor value	numerical display
video feed	rectangular display

Table 2.1 – Parameters type detected from the Smart Device metadata and UI elements assignments

Last, the teacher has to choose an educational platform in which the generated web app will be embedded (See Figure 2.3). The automatic web app generator provides apps which can be embedded in Graasp or in an LTI consumer platform (such as Moodle, or edX). Both types of web apps have the same user interface and provide a similar user experience: the resulting lab client application will automatically instantiate a WebSocket connection with the Smart Device whenever a user accesses the lab client, update the UI components of the sensors upon receiving new sensor values, handle the actuator changes performed by the user and send the new actuator data to the Smart Device. According to the teacher's selection of one or more experiments, the application will contain one or more tabs. Each tab represents a selected experimental configuration. Clicking on a tab in the client application will result in accessing the corresponding experiment, and displaying all the sensors and actuators associated with that setup. Next we detail how the apps destined for LTI-consumer platform and Graasp are embedded in the respective environments.

2.4. Validating and Evaluating the Smart Device Configurations with the Automatic Web App Generator

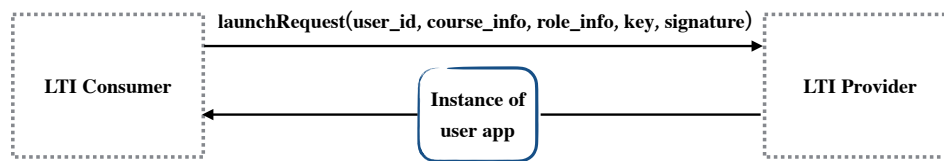


Figure 2.5 – LTI consumer and provider communication

2.4.1 Integration in LTI-Consumer Platforms

LTI stands for Learning Tools Interoperability and is a standard developed by IMS² to allow third-party applications to be embedded in LTI-consumer platforms such as Moodle, edX, Blackboard or other. In their terms, a *tool consumer* is the platform and the *provider* is the external tool. Figure 2.5 shows how an LTI tool consumer and provider communicate.

The basic procedure for using LTI starts when the instructor or LMS platform administrator asks and gains access to a third-party learning tool (i.e. a tool offered by an external entity). The tool's administrator provides the LMS administrator or instructor a *URL*, *key*, and *secret* for that tool. The external tool receives a launch request (denoted with *launchRequest* in Figure 2.5) that includes user identity (*user_id*), course information (*course_info*), role information (*role_info*), and the key (*key*) and signature (*signature*). The launch information is sent using an HTTP form, with the LTI data elements in hidden form fields and automatically submitted to the external tool using JavaScript. The data in the HTTP form is signed using the OAuth³ security standard so the external tool can be assured that the launch data was not modified between the time the LMS generated and signed the data, and the time that the tool received the data. Once the launch request is received, the tool either redirects the user's browser to some other URL, or it renders the requested user interface straight-away.

Even though LTI is a specification, certain platforms have specific implementations [24], hence limiting the possibility of generating a directly-ready-to-use LTI provider. Additionally, a tool provider is supposed to implement the request-response protocol explained above. Technically, it is possible to automatically generate the code implementing the protocol, but not as a deployed application.

The automatic web app generator provides teachers with the first building block for integrating the lab in such platforms: an html file which can be used to configure the lab's integration through the teacher's tools in the target platform [49], hence not limiting teachers to one LTI consumer platform or the other. In that case, if teachers don't get special assistance for augmenting the HTML version of the web app with LTI-specific mechanisms, the basic integration in the target platform is possible, where the web app can be used in the platform, but does not exchange context and information.

²<http://www.imsglobal.org/activity/learning-tools-interoperability>

³www.oauth.net

2.4.2 Integration in Graasp and Interoperability with the GoLabz Infrastructure

If Graasp is chosen as the target educational platform, then the generated lab client application is an OpenSocial widget which can be embedded in the platform. Consequently, the OpenSocial widget implements all the functionalities of the LTI-targeted application mentioned in the previous Section, in addition to the following features which allow it to be interoperable with other Go-Lab tools:

- Action logging: the generated Graasp gadget uses the ActionLogger library⁴, which provides an easy mechanism for logging the activities of the students. Interactions with the different UI components of the web app are saved as Activity Streams that have the actor-verb-object format. The logged activities can later be used to perform learning analytics.
- Saving experimental data: the lab application allows students to save the actuator and sensor data that were acquired while conducting the experiment. The data is saved in a specific format, that allows students to use it in other applications on the platform. For example, the students can have a graphical view of the experimental results using the Data Viewer application⁵, as also shown in Figure 1.6.

2.5 Proof-of-concept Example

Starting from the basic interferometry device (detailed in Section 1.6.1), the teachers at Gymnase de Morges designed a series of experiments to study different characteristics of light interference, in both the classical and quantum modes. They added a number of components to the basic MZI presented before. Figure 2.6 depicts the layout of the MZI designed by the teachers. They added a density filter denoted DF in the figure to switch between low-intensity and high-intensity experimentation (or between classical to quantum modes), the opaque shutters S1 and S2, the piezo actuator (PA) mounted on mirror M2 and the photodiode (PD) mounted on D1. The arrows in the figure indicate the direction in which the components can be controlled. More details about the experiments is provided in later sections of the thesis to better contextualize the information.

The mindmap in Figure 2.7 shows two possible experiments that can be done with the Mach-Zehnder interferometer, upon which we base our explanation of the implementation and function of the automatic web app generator presented in Section 2.4.

⁴<https://github.com/go-lab/ils/wiki/ActionLogger>

⁵<http://go-lab.gw.utwente.nl/production/dataViewer/build/dataViewerTool.xml>

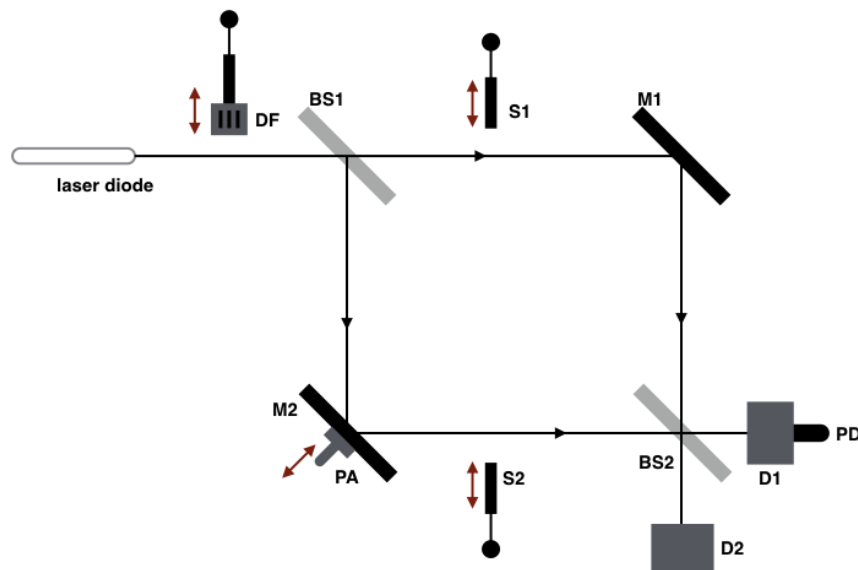


Figure 2.6 – Mach Zehnder Interferometer layout design by teachers at Gymnase de Morges.

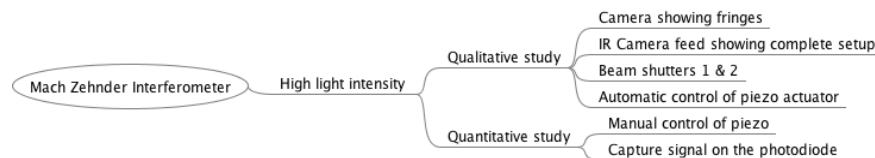


Figure 2.7 – Mindmap of two possible Mach-Zehnder experiments

The first and second experiments are conducted in a high light intensity setup, meaning that the density filter is not attenuating the intensity of the light coming from the monochromatic light beam. The first experiment enables the users to qualitatively understand light interference, by visualizing the resulting fringes on the screen, and/or also the feed from the infrared camera, in addition to depicting the direction in which the fringes move when the mirror mounted with a piezo actuator manually controlled with a voltage which is increasing or decreasing in value. In the second experiment, the students can quantitatively study light interference by observing the emitted signal from the photodiode as the piezo is controlled with a triangular signal causing a translation motion.

Chapter 2. The Smart Device Configurations

How is it translated to the configurations in the metadata?

When a *getConfigurations* api call is sent to the Smart Device hosting this laboratory, the following response is received:

```
{ "method": "getConfigurations",
  "experiments": [{
    "configurationId": "qualitative",
    "fullName": "Qualitative Study",
    "description": "Observing light interference on the screen",
    "sensors": [{"sensorId": "Video"}, {"sensorId": "VideoIR"}],
    "actuators": [{"actuatorId": "laser"}, {"actuatorId": "piezo"},
    {"actuatorId": "bs1"}, {"actuatorId": "bs2"}]
  }, {
    "configurationId": "quantitative",
    "fullName": "Quantitative Study",
    "description": "Studying the signal provided by the photodiode",
    "sensors": [{"sensorId": "photodiode"}],
    "actuators": [{"actuatorId": "laser"}, {"actuatorId": "piezo"},
    {"actuatorId": "bs1"}, {"actuatorId": "bs2"}]
  }
]}
```

The response shows that there are two possible experiments with the *configurationIds* “qualitative” and “quantitative”. Accordingly, the list of sensors and actuators for each of the experiments can be either used from this response.

Teacher Customization

The automatic web app generator provides a basic and fully functional client application for operating a cyber-physical lab. The components of the user interface are very basic and might not be visually attractive. Using the generated code, the teachers can further personalize the UI appearance to their taste and needs. For example, a teacher from the Gymnase de Morges in Switzerland chose to customize the UI to be embedded in Graasp as shown in Figure 1.5 in Chapter 1.

In this widget, there are two tabs to switch between two possible experiments. In the Quantitative Study tab, there is a simulation diagram which allows students to control the lab by clicking on the corresponding image of a component. For example, to turn the laser beam ON/OFF it is enough to click on the box representing the light source. On the diagram are also present the placements of the IR camera and the normal camera allowing the student to know about the perspective of the video feeds. In this widget, the teacher chose to only display the video coming from Camera 2 showing the fringes on the screen. Next to it is a graphing tool that shows the signal captured by the photo diode in real-time. Since the teacher doesn’t want the students to have to scroll, and since the simulation diagram conveys a real-time status of the lab, they decided that there are enough UI components for the students to conduct the experiment while having a good user experience.

Of course, the UI could have been customized otherwise to show the UI components differently, or to resize them in a different way. For example, an input box to control the piezo actuator could have been a replacement for the slider control. Also, instead of only showing the feed of Camera 2, both feeds from Camera 1 and Camera 2 could have been shown, in addition to the graphing tool. All of this is possible by starting from the code provided by the automatic web app generator. The proposed solution alleviates the burden of establishing connections and parsing the CPL API, making it more easy to personalize the appearance of user client according to a desired user experience.

2.6 Discussion

Solutions such as the proposed in [59] provided ground to support the *personalization* of configurable CPL web apps through the disaggregation of the main app into smaller apps, which give access to separate lab components (check Section 2.1). Teachers can make a selection of the available mini-apps to put together into a new CPL web app. A major drawback of such an approach is the dependency that still exists between the CPL mini web apps and the server. Even through the access to the components of the configurable CPL is disaggregated, it is only possible through the provided web apps, which implement the proprietary technologies of the system. In this case, *portability* and *interoperability* are not supported.

In [63], limitations such as CPL web app *portability* and *interoperability* are addressed by resorting to adapting API design principles, and proposing an implementation of CPLs based on software oriented computing principles [70]. The resulting Smart Device specification describes the CPL server through an API, theoretically allowing the automatic generation of CPL web apps, by separately calling desired CPL services (sensors and actuators), and generating CPL web apps independently from the technologies adopted for the CPL server.

The adaptation of the EjsS tool to generate web apps for CPLs rather than simulations [6, 7, 18] allows the automatic generation web apps and anticipates the need to generate new apps for each newly added CPL to the considered platform. While the solution provides teacher with full autonomy to personalize the apps, the framework is proprietary to the implementation of the JIL server (all details discussed in Section 2.1).

Recall the research question **RQ1: how can we support teachers in the generation of web applications for configurable cyber-physical labs, without the mediation of an application developer and the lab provider?** For both solutions in [59] and [63], if they are provided to teachers in an accessible user experience, the generation of CPL web apps is possible. But without knowledge of how a system's components are related and dependent, an "experimentally meaningful" selection is not possible (a number of components could be aggregated without resulting in a pedagogically meaningful experiment). Teachers are still dependent on the lab provider to inquire about the possible **configurations**.

To fully support the automatic generation of CPL web apps, we proposed in this chapter an extension to the Smart Device Specification [63], by providing a model of an experimental configuration and corresponding API calls (**contribution 1a**). When the CPL server provides such API calls, an automatic web app generator can be implemented and provided for teachers to use. We proposed an implementation of an automatic web app generator which can be used with any CPL implementing the extended Smart Device Specification (**contribution 1b**). The proposed CPL web app generator provides 2 possible versions of the generated web app, both having the same user interface, but different support for *portability* and *interoperability*. The first version provides full integration with the target embedding platform (Graasp) by implementing mechanisms for context retrieval and saving experimental data. In this case, *interoperability* is supported, but *portability* is not due to the specific integration technology used (OpenSocial presented in Section 1.2.1). The second version is in HTML, providing the possibility to integrate in a wild array of platforms such as LTI platforms. In this case, *portability* is supported but *interoperability* with the hosting platform is not. In Section 2.4.1, we discussed in detail the specifics of the LTI specification which hinder the possibility of automatically generating all that is needed for the integration in platforms depending on that specification.

2.7 Conclusion

To describe configurable CPLs and enable the automatic generation of interfacing web apps, it is not enough to solely rely on the description of the individual services making up the CPL. In this chapter, we presented the extended Smart Device Specification to support the description of CPLs supporting different configurations or experiments (**contribution 1a**). This extension further enables the automatic generation of user clients by specifying the relationships between the lab components in a configuration. We also proposed a web app generator tool, which helps teachers in autonomously creating client applications for different target platforms: Graasp or an LTI-consumer environments (**contribution 1b**). The tool implements different levels of integration, depending on the embedding platforms. The web app generator is openly shared under the CC-BY-NC⁶ creative commons licenses on a public repository⁷. Then, we showed how the proposed extension and tool are used with an example CPL for the generation of the basic web app, which is later customized for the specific expectations of a teacher.

The work presented in this chapter is part of the in-progress IEEE standard for Networked Smart Learning Objects for Online Laboratories⁸, part of the P1876 Working Group (NSLOL WG).

⁶<https://creativecommons.org/licenses/by-nc/2.0/>

⁷http://shindig2.epfl.ch/gadget/automatic_gadget_generator/

⁸https://standards.ieee.org/email/2012_09_cfp_P1876wg_web.html

3 Activity Tracking Infrastructure for Embedded Cyber-Physical Labs

The dynamics in a classroom or lab room contribute to the effectiveness of the learning experience. For instance, the physical presence amongst classmates facilitates many self-regulating mechanisms. In the lab room, **students** experiment and for many reasons they might want to communicate. For instance a group might figure that they are taking too much time to make something work, so they check with their classmates to compare steps taken so far, and try to debug. Another example is getting results which don't match with the theoretically calculated values, and resorting to checking with another group their results. Students walking around the lab room asking about the status of their classmates, sharing their data and having their questions answered is based on the availability of a data flow, which students can observe and reflect upon and take steps to adjust their procedures [20]. In the physical setting, where the lab setups are used by students in-person, the lab technician is able to identify why equipment is breaking down. They are either present while the students are using the labs, or keep track of which experiments were conducted with the equipment they are maintaining. They know which group of students came in the lab room, which group used which setup, and other indicators which would help them debug and maintain the equipment. Thinking about the students connecting to a particular CPL as the group of students doing the same experiments with a lab, and about the **lab owner** as the lab technician for CPLs, supporting both **students** and **lab owners** to have a similar awareness in online learning environments is necessary.

Through interacting with cyber-physical labs, learners generate data which is valuable to them, their classmates, teachers and lab owners. While all parties involved in the CPL experimentation experience are interested in the generated data, they each have different needs and concerns regarding data collection and analysis. As teachers are usually well-supported with data collection and reporting in online learning platforms, and more specifically for both Graasp and edX considered in this thesis, we are interested in addressing the needs of **students** and **lab owners**.

Chapter 3. Activity Tracking Infrastructure for Embedded Cyber-Physical Labs

In this chapter we aim to answer the following research questions:

RQ2: What are the experience data that students and lab owners are interested in?

RQ3: What are students concerned about in their online privacy?

RQ4: How to design an activity tracking infrastructure which responds to the needs of both students and lab owners, while preserving students' privacy?

This chapter is structured as follows: first we provide an overview of related work for activity tracking infrastructures. Second, we study the needs and concerns of students and lab owners from the interaction data through questionnaires (**contributions 2 and 3** answering **RQ2** and **RQ3**). Third, we propose a vocabulary to format the activity traces to record (**contribution 4a**). Then, we devise an architectural model of the activity tracking infrastructure (**contribution 4b**). Hence, the proposed activity tracking infrastructure is composed of a vocabulary and an architectural model responding to **RQ4**. Last, we provide two implementation examples of the proposed infrastructure, and evaluate one of them in real-settings.

3.1 Related Work

In this section, we overview related work to three aspects of activity tracking in online learning environments: the architecture pertaining to how the data is collected and archived, the data format adopted in formatting the traces, and privacy management modalities.

3.1.1 Activity Tracking Architectures in Online Learning Environments

A number of activity tracking frameworks which can be utilized for educational settings exist. For example, there are the “all-purpose” solutions such as Google Analytics¹ (GA). GA provides rich APIs to save the traces to Google’s servers, which can be then visualized on the Google Analytics platform. The data can then be manually exported into various useful formats, or programmatically through the Analytics Reporting API². GA represents two limitations for our purposes: first although the APIs are flexible, they limit the use of fields for saving the traces leads to losing granularity in describing activity, leading to limitations in studying the data. Second, many privacy problems related to governmental and institutional guidelines arise, due to the fact that the data is saved on the Google servers.

In [40], the authors propose a “flexible” and “extendable” learning analytics³ infrastructure. Their design is based on 3 requirements: action logging, user feedback, and ex-post analysis. This respectively means that a user can save and retrieve data related to their activity on a web-based learning platform, can receive feedback from the infrastructure for guidance, and teachers can

¹<https://developers.google.com/analytics/>

²<https://developers.google.com/analytics/devguides/reporting/core/v4/>

³Learning Analytics is the collection and analysis of learner’s data, for the purposes of understanding and optimizing learning and the contexts where they occur.

utilize the collected data over time to study learning traces at scale. While their approach supports three important services of a learning analytics infrastructure, the collected data can only be used in proprietary analysis tools.

In [76] and [77], the authors present a contextual solution to save students tracks. The architecture enables teachers to choose whether they want to track students online or not, and accordingly can then use awareness and reflection tools to pull user data in a specific context (online lesson), and visualize them in dashboards. While this solution contextualizes learning analytics, it doesn't enable students to get their data unless the teacher chooses to. This solution is designed to mainly support teachers in understanding the dynamics of an online class (example dashboards shown in Section 1.3, Figure 1.11). With this solution, the learning traces are saved in the embedding platform, and only accessible through it to enforce privacy.

3.1.2 Activity Tracking Data Formats in Online Learning Environments

There have been efforts expanded by educational software providers and learning analytics advocates to standardize educational data formats. At the present time, the most prominent are the Caliper standard developed by the IMS Global Learning Consortium ⁴ and the xAPI specification by ADL ⁵ (Advanced Distributed Learning).

The Caliper standard [11] defines an API (the Sensor API) for capturing student interaction, and provides a rich data structure for capturing user interaction, storing and retrieving it. The data format is based on the *actor-verb-object* format, and is flexible to hold more elements. The data structure can carry a lot of information, and supports information contextualization. The data formatted according the Caliper specification can be consumed by content repositories, reporting tools or LA (Learning Analytics) tools which are conformant with the specification. The community has provided a software library which implements the Sensor API, called *Caliper Sensor*, and an implementation of a data storage system compatible with the specification referred to as the *EventStore*.

The xAPI specification [47] previously known as TinCan is also based on the *actor-verb-object* format, but provides an ampler formalization of additional parameters such as *context*, *activity* and others. The specification also defines a set of guidelines for formatting tracking data for a given learning experience, for example watching a video or attending a conference. The guidelines are expected to ensure consistency in describing the same learning experience, by different data trackers. In other words, watching a video should be described the same way in edX and Graasp for example. A data storage system which implements the specification is referred to as an LRS (Learning Record Store). ADL and other contributors such as the H2Labs⁶ and Rustici Software⁷ provide rich software and libraries to make full use of the specification.

⁴<https://www.imsglobal.org/>

⁵<https://www.adlnet.gov>

⁶<https://www.ht2labs.com/>

⁷<https://rusticissoftware.com/>

3.1.3 Privacy in Online Learning Environments

Privacy in educational online environments is managed differently depending on the application being used, the country from which it is hosted, the organization providing it, and the target audiences. Online users are not only adults, often children and teenagers are users of websites, and fall into an age group which may not fully understand and appreciate privacy concerns and implications. Hence, in these cases, an adult being a parent or a teacher is responsible for ensuring online user privacy preservation. Other than regulations which define who is responsible for what, according to Boyd et al. [3] and Lessig [38] the software architecture of systems plays a principle role in regulating data privacy in digital environments. They claim that privacy concerns in online environments emanate from four affordances of networked technologies:

1. Persistence: digital data are automatically collected and saved
2. Replicability: digital content is easily duplicated
3. Scalability: the potential of spreading the digital content is great
4. Searchability: digital content is often found through search engines (may it be global to the web such as google, or local to a platform)

In [77], the authors propose an agent-based approach to handle privacy management in online learning contexts where students are minors. In their approach, teachers have full control and responsibility for managing students' data. The agent-based approach is claimed to be a flexible, contextual, and intuitive solution. This privacy management approach abstracts a commonly perceived monitoring paradigm: if a person is in a room, they can monitor what is happening in that room; if they are not, they cannot. While this approach gives the teacher full control of activity tracking it presents three problems when students are adults who can make sound judgments about their privacy. First, the students have no idea if they are tracked or not, unless the teacher states it for them. Second, the students and the teacher don't know what is being tracked, they can only see the activity graphs. And last, there is no clear explanation on how the data is saved and where, and who is using it for what.

As a matter of fact, to the extent of our knowledge, and as mentioned in Section 1.3.2, current online learning platforms don't provide users with a data privacy management choice. For example in edX it's the system administrator who decides if tracking is activated and in Graasp it is the teacher.

3.2 Requirements Elicitation

To answer the research questions formulated in this chapter, we start by eliciting the needs of students and lab owners as stakeholders of the data generated when using a CPL embedded in an online learning environment, and as part of a learning scenario. In this section, we present and analyze two questionnaires sent to students and one to lab owners.

Note on the visualizations in this chapter: the type of graph used to illustrate the responses to a question correspond to the type of answers. Pie charts correspond one-choice questions, stacked bar charts correspond to multiple choice questions, and boxplots correspond to rating questions, where they represent the distribution of answers.

3.2.1 Students Questionnaires

We send out a questionnaire⁸ to the students of the Control Systems and Automatic Control courses at EPFL for the academic years 2015–2016 and 2016–2017. All of the students have used the MOOC for the completion of the lab requirements of either courses, and they were familiar with the lab and the online platform. The questionnaire was sent to 282 students, and the total number of respondents is 33 (11.7% response rate). We consider any answer with more than 50% of contribution from respondents as the answer which has the agreement of the majority on the corresponding question.

Questionnaire 1: Tracking and Sharing Data

The questionnaire was designed to answer three main questions:

1. How important it is to the students to evaluate their own progress while doing lab work, and compare it to the rest of the class?
2. What is the student willing to share with other classmates regarding their activity in the lab session, so others can compare their performance to theirs?
3. What would the students want to be able to do with their experimental results?

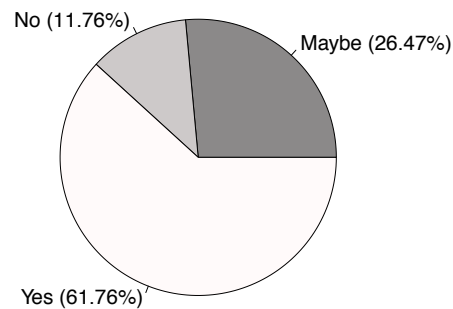
⁸<https://goo.gl/WhghKG>

Chapter 3. Activity Tracking Infrastructure for Embedded Cyber-Physical Labs

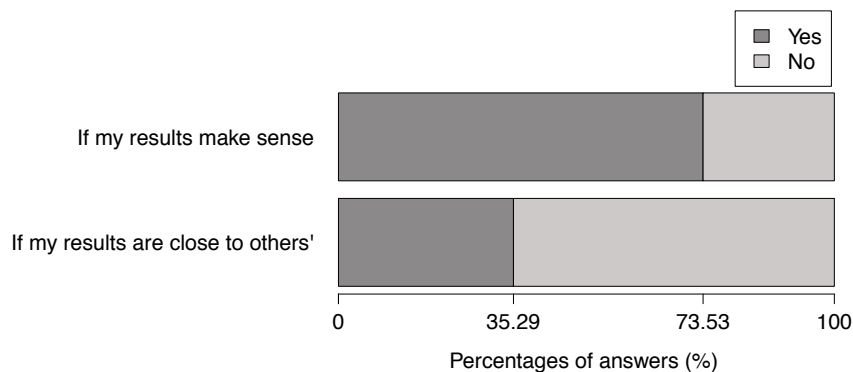
Part 1: Awareness of peers in the lab

The aim of this part of the questionnaire is to understand what interests students in the dynamics of a lab session: progress of classmates, how they know if they are on the right track, what kind of problems they face, how they solve them when they are co-located in the lab with peers and teaching assistants. This part has 9 questions detailed hereafter with their corresponding answers:

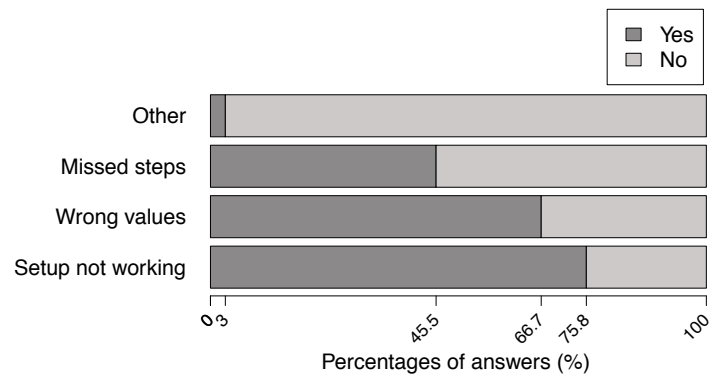
Q1: Are you interested in the progress of your classmates in the lab session?



Q2: How do you know that you are on the right track in your experiments?

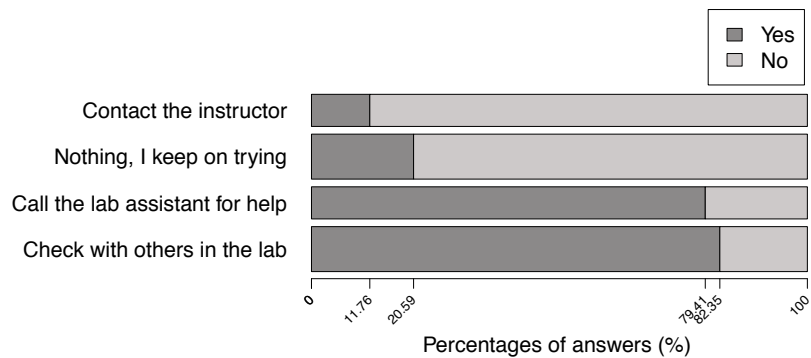


Q3: What kind of problems do you face when doing an experiment in the lab?



3.2. Requirements Elicitation

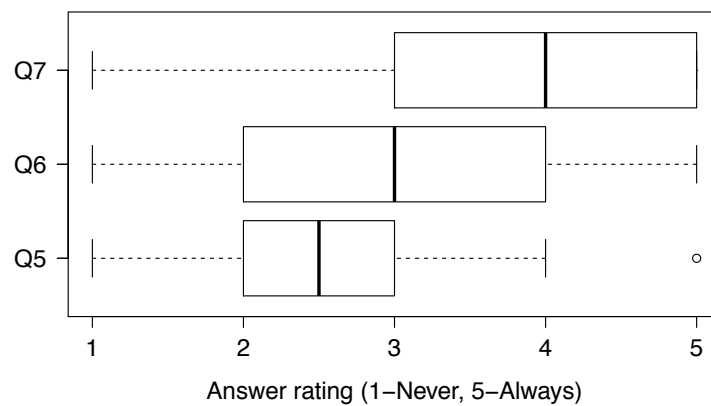
Q4: If you are in the lab, and having trouble with your experiments, what do you do?



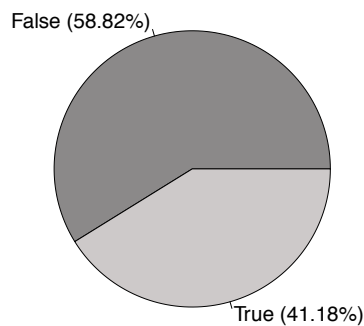
Q5: Whether you need help or not, how often do you check with others their steps in the experimentation protocol, regardless of being in need for help?

Q6: Whether you need help or not, how often do you check with others their experimentation results?

Q7: How often do you discuss your results with others in the lab?

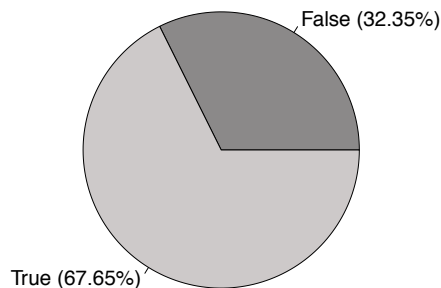


Q8: I only check with others their results if I think my results don't make sense.



Chapter 3. Activity Tracking Infrastructure for Embedded Cyber-Physical Labs

Q9: I only check with others their steps in the experimentation protocol if I think I did something wrong.



Summary

The majority of the responding students say that they are interested in knowing the progress of others in the class (Q1). To check the correctness of their work, around 74% rely on their own judgment of expected correct values, and around 34% compare their results to others' (Q2). The most encountered problem is equipment malfunction (75.8%), followed by getting wrong values (66.7%), then missing steps (45.5%), and last 3% for other miscellaneous issues (Q3). To identify the cause of the trouble they are in, 82.8% of the responders check with other classmates what could have gone wrong, or with the teaching assistants present in the lab (79.41%). 20.59% try to identify and solve their problems on their own without any intervention, and 11.76% contact the instructor of the lab (Q4). From the answers to Q5 and Q6, we see that students tend to check the experimental values with others regardless of whether they are stuck or not, but less likely the steps taken so far (Q5 & Q6). The answers to Q7 show that students tend to discuss with each other their results. From Q8 & Q9 we see that 41.18% of the respondents only check their numbers with others if they think they don't make sense, while the other 58.82% double checks anyway. Around 67% of the respondents check the steps if they think they did something wrong, and only around 33% check anyway.

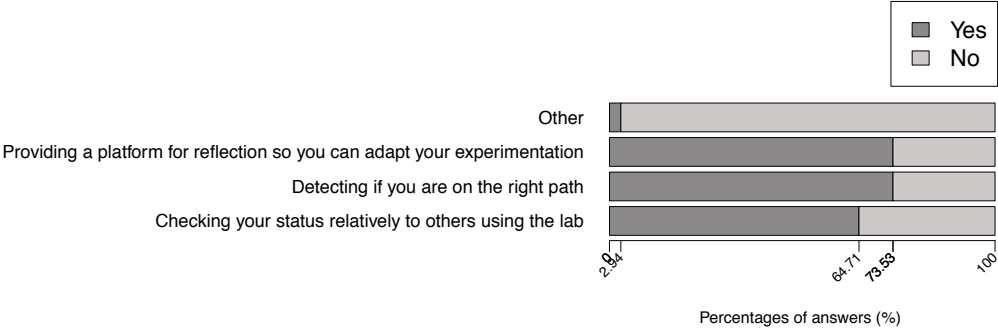
Part 2: What to expect in an activity dashboard

We assume that students are familiar with the concept of 'activity dashboards' given the ubiquitous use of smart phones, smart watches, and other smart devices to track and report on activities such as sport performance or workout completion. We expect that the students will be able to project the same concepts to the activities happening in a lab session. But to be sure that all students understand the same thing regarding the activity dashboards we are referring to, we include this description for the second section of the questionnaire:

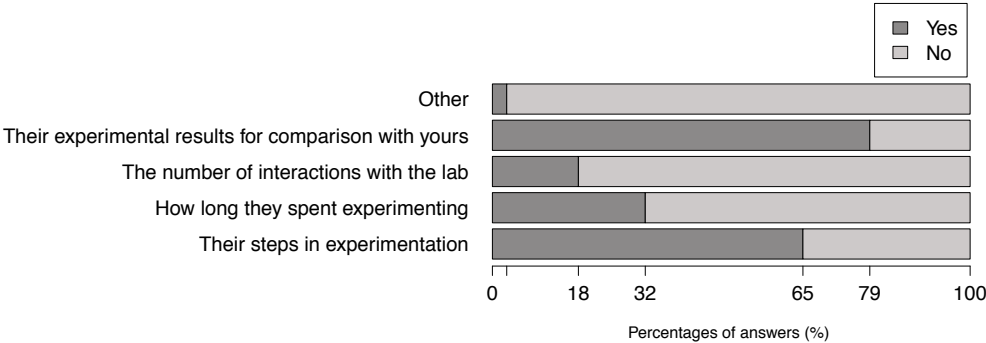
"Activity dashboards are data visualizations (e.g. graphs, plots...) which show you different metrics representative of the current activities in a medium. An example for an activity dashboard visualization in the lab would be the max, min, and average time spent in a given lab session. Another example is the most popular sequence of steps in the protocol..."

This section comprises 3 questions:

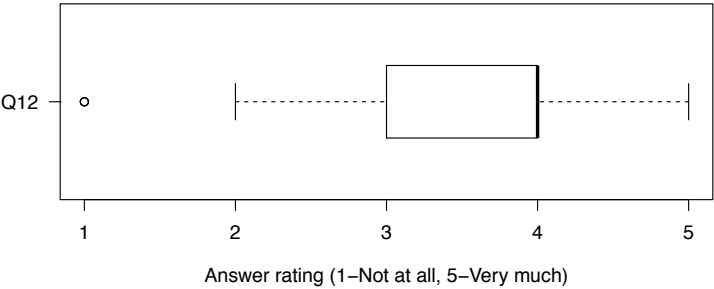
Q10: If you are doing your lab assignment online, away from the lab room, do you think such dashboards will help you in?



Q11: What would you like to know about others in the lab?



Q12: How useful those dashboards would be?



Chapter 3. Activity Tracking Infrastructure for Embedded Cyber-Physical Labs

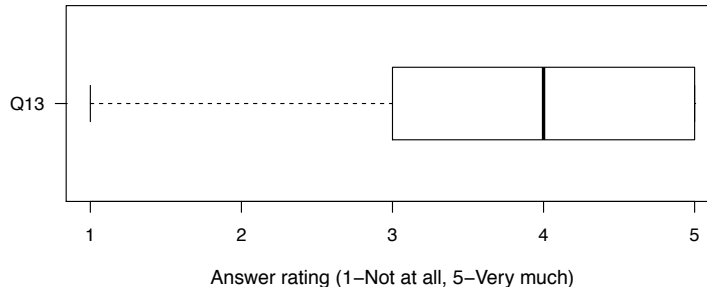
Summary

73.53% (Q10) of the respondents think that activity dashboards provide a platform for awareness and reflection to self-regulate. 64.71% believe it would help them compare their progress to others, while 2.94% (1 respondent) thought they provide nothing interesting. From Q11, we see that students are mostly interested in the experimental results of others (79%), then by the steps of their colleagues (65%), followed by the time spent experimenting (32%), the number of interactions with the lab (18%), and 3% nothing. Last in response to Q12, students tend to find activity dashboards useful.

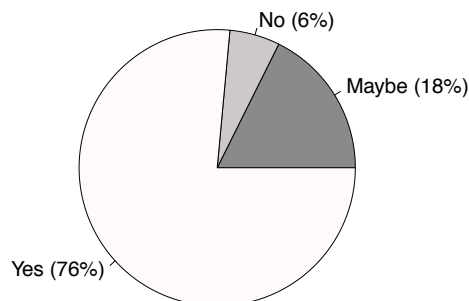
Part 3: Experimental Results

In this part of the questionnaire, we want to know what students would do with their CPL experimentation measurements. In the MOOC, they can save their results in the context of the course and retrieve them in another tool for interactive graphing, but cannot save them for their personal use (as presented in Section 1.3). Additionally, we want to know if they are interested in linking the measurements to a sequence of actions. We provide them with our definition of ‘sequence’: “when we say **sequence** we mean the steps you have taken leading to your results (*adjusting parameters, activating/deactivating a component in the lab...*)”, to make sure they all have a clear idea. This part has 8 questions:

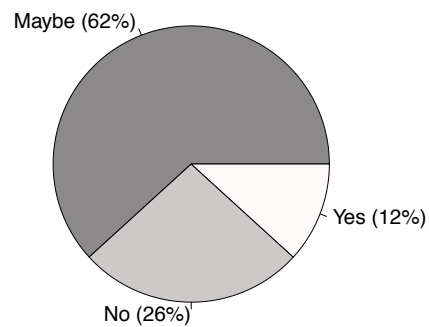
Q13: How convenient was it to be able to save and retrieve your experimentation results between tools in the MOOC (i.e popup menu for selecting the asset)?



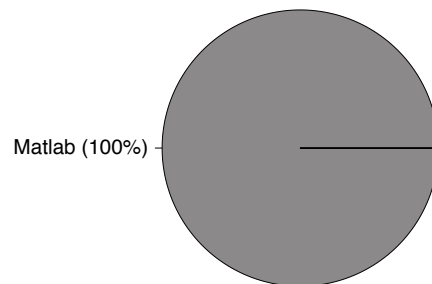
Q14: Would have liked to be able to get/save your experimental data for your own records?



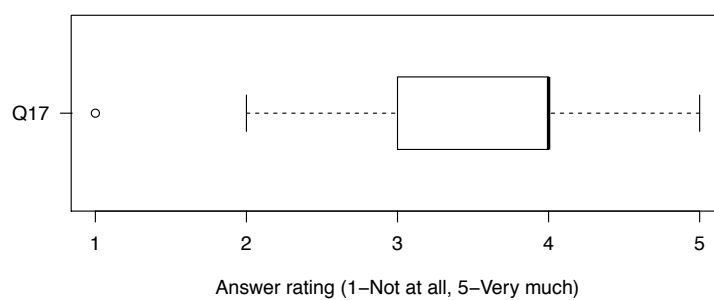
Q15: Would you have used your data in other tools than the ones provided in the MOOC?



Q16: If yes, in which tool(s) would you use your data?

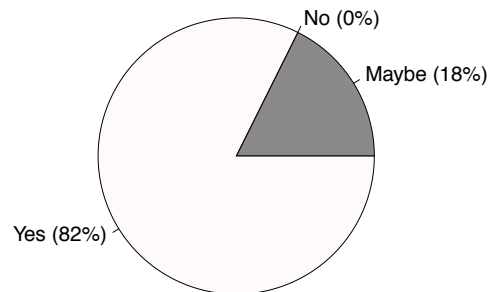


Q17: How useful do you think it is, to have your data linked to a sequence of steps in your experimentation protocol? For e.g. to be able to replay your steps in a simulation, or verifying your results.

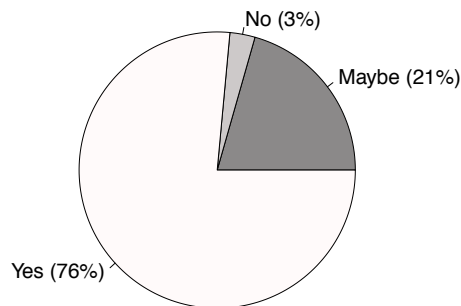


Chapter 3. Activity Tracking Infrastructure for Embedded Cyber-Physical Labs

Q18: Are you willing to share with your classmates your sequence and data if they ask for help?



Q19: Would you ask a classmate for their sequence and data if you are running into trouble?



Summary

Students tend to find the tool to save and retrieve their experimental results very useful (Q13). 76% of the respondents said they want to be able to save their results for their own records, 18% are not sure, and 6% said no (Q14). From Q15, we see that students are not sure whether they would use their measurements in tools other than the one provided to them in the MOOC. This is at once an indicator that perhaps they are not aware of other tools, or/and that they find the Sysquake tool enough for the purpose of the course. Sysquake is a tool similar to Matlab, it accepts all Matlab commands, with an added value of letting users interact with the graphs, and is integrated in the MOOC using its web version. From Q17, we see that students tend to believe that it is useful to have their actions linked to measurements. The vast majority of the respondents are willing to share their sequence and data with other classmates if asked for help (82%), 18% are not sure, but no one said no. 76% of the respondents would ask for help if they were stuck, 21% are not sure they would, and 3% said no (Q19).

To Q20, which is an open-ended question (is there any other information you would like to share, or suggestion you would like to leave?), two respondents had a take on privacy and collaboration:

Answer 1: *'I think sharing the results could be very usefull, but I wouldn't share it with all my classmates. I usually compare it with my friends and those who needed them, but I must say that I don't want to share them with everybody, since even those who did nothing to complete the tasks could also take advantage of my work.'*

Answer 2: *'I think collaboration between students should be left to individual appreciation'*

Questionnaire 2: Privacy

In light of the comments left from 2 students regarding sharing preferences and in regards to the analysis of privacy concerns in Section 3.1, we sent another questionnaire ⁹ to the same pool of students of the previous questionnaire. We wanted to understand how they feel about their privacy when they are tracked, and how their data is shared, what are their concerns, and what would make them trust the system and share their data. The questionnaire is made of 8 questions. Hereafter we detail them with their respective answers:

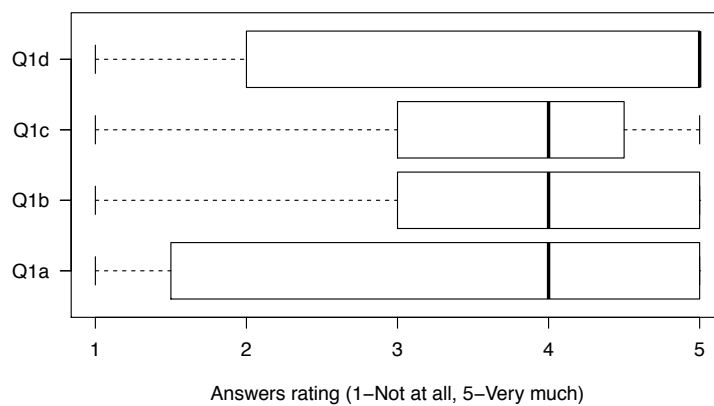
Q1: I am concerned about my privacy regarding:

Q1a : Disclosing my identity

Q1b : Linking the sequence to me

Q1c : Linking the experimental results to me

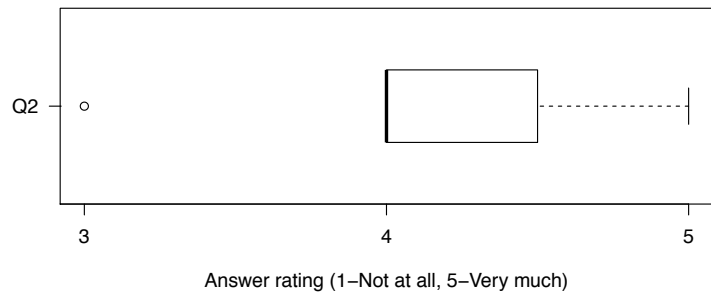
Q1d : Sharing my identity with class-outsiders



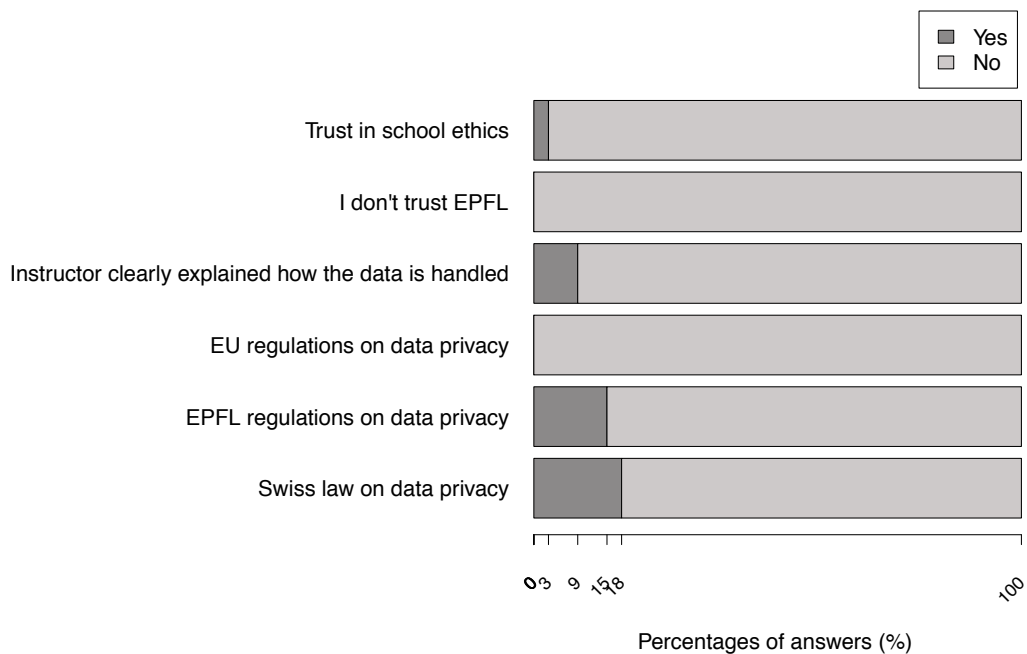
⁹<https://goo.gl/fPPY9m>

Chapter 3. Activity Tracking Infrastructure for Embedded Cyber-Physical Labs

Q2: I trust that EPFL appropriately regulates my data privacy (i.e. doesn't share them with outsiders, will ask my permission for using the data, will anonymise the data...)



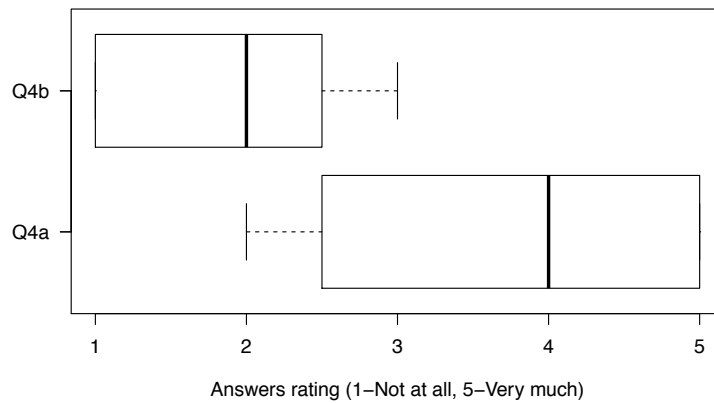
Q3: I trust EPFL with my data because of:



Q4: I am okay sharing my sequence and experimental data:

Q4a : If it's anonymous

Q4b : Even if it's not anonymous

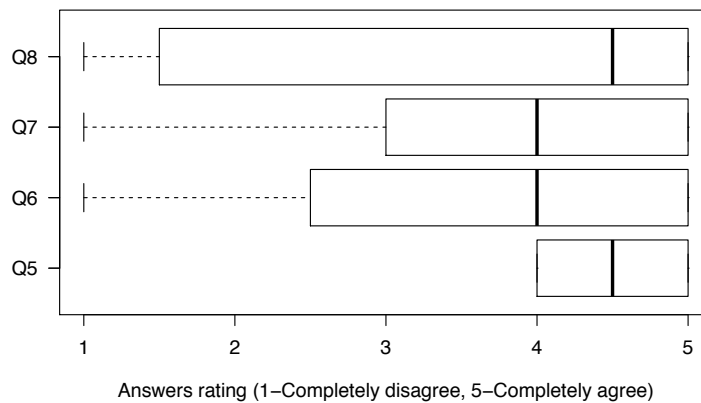


Q5: I think those who don't share their sequence and results should not be able to see those dashboards.

Q6: I will share my identity with my data only with my classmates who shared their identity (i.e. i accept to link my identity to my data and show it only to those who did the same).

Q7: Even if I am willing to share my sequence and results, I prefer to have an opt-out option.

Q8: If the only way I can see the dashboards is by sharing my sequence and results, I will share them.



Chapter 3. Activity Tracking Infrastructure for Embedded Cyber-Physical Labs

Summary

From the first question of this questionnaire, students show comparable concerns regarding disclosing their identity, linking the activity sequences to them and linking the experimental results to them with all medians equal to 4 (Q1a, Q1b and Q1c); but they show a higher concern regarding disclosing their identity to class-outsiders, with a median of 5 (Q1d in the same figure). From Q2, we can see that students do trust EPFL in regulating their data privacy (a median of 4 with a tendency to higher ratings), aside from one outlier with a fair trust rated at 3 out of 5. Their main reasons why students trust EPFL with their data are in order the Swiss law on data privacy, EPFL regulation on data privacy, and because the course instructor explained to them what will be done with their data (Q3). Q4 reveals that students would be uncomfortable if their data was not anonymized prior to sharing (Q4b with a median of 2 and answers tending to lower ratings). From Q5, it is clear that students are biased towards the merit of having access to the dashboards by contribution to them (i.e. allowing the system to track them), the median is 4.5 and the distribution of answers is contained between 4 and 5. Sharing their identity and data with those who did share theirs seems to encourage students to do it (Q6). Even though from Q7 we see that students are willing to share their traces, they prefer to have the option of disabling the tracking. From Q8, we see that if the only way to get the dashboards is by activating own's tracking, students' ratings are spread between complete agreement and disagreement, with a concentration near complete agreement (median around 4.5).

3.2.2 Lab Owners Questionnaire

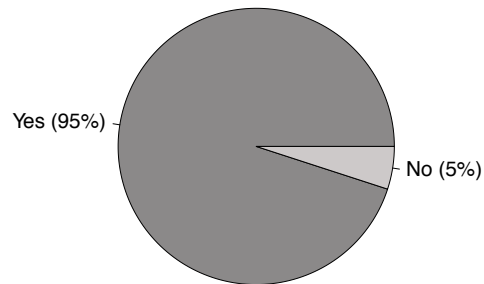
We design and send out a questionnaire¹⁰ to lab owners. In order to understand lab owners' monitoring needs for cyber-physical labs, we contacted 80 lab owners, 20 responded (25% response rate). The questionnaire has 3 parts: the first to find out what lab owners currently monitor and for which purposes, the second to check their interest in advertising for their labs using the monitoring data they collect, and the third is to gather lab owners' demographics.

¹⁰<https://goo.gl/fLWUIU>

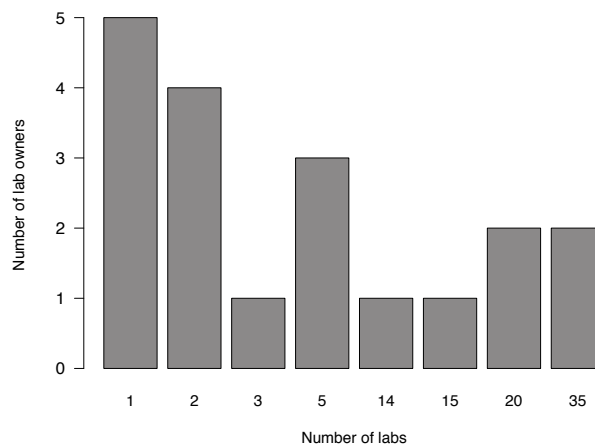
Part 1: Monitoring Activity

In this part, we want to know what currently lab owners track. There are 5 questions, detailed hereafter:

Q1: Do you monitor your labs?

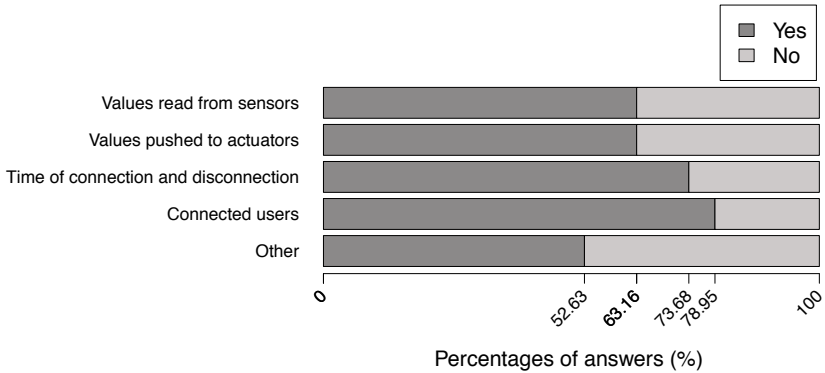


Q2: How many labs do you have?



Chapter 3. Activity Tracking Infrastructure for Embedded Cyber-Physical Labs

Q3: What do you monitor?



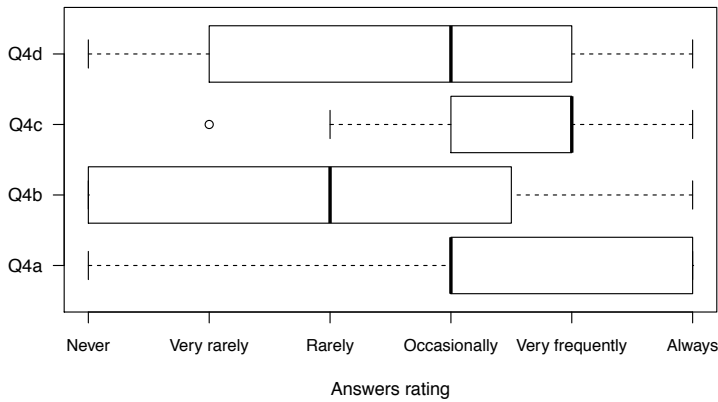
Q4: Do you monitor for:

Q4a : Statistics of lab usage

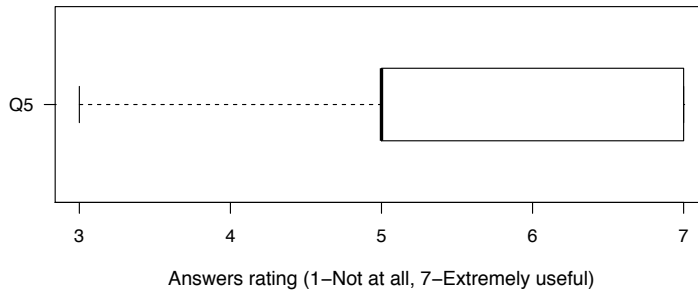
Q4b : System modeling

Q4c : Failure/Security auditing

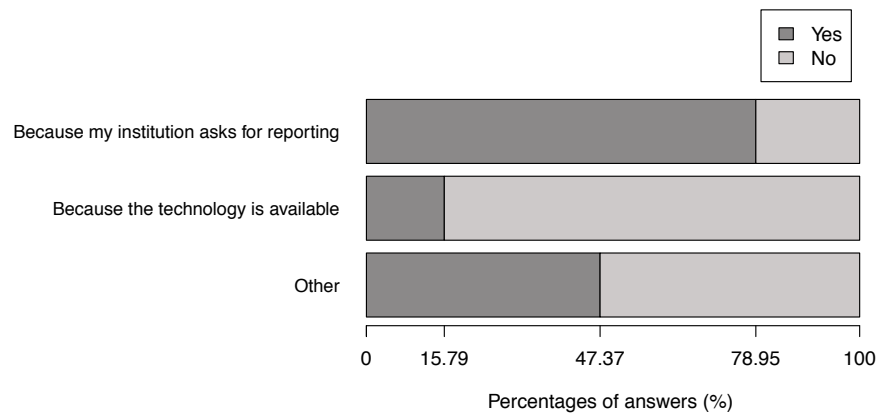
Q4d : Load balancing



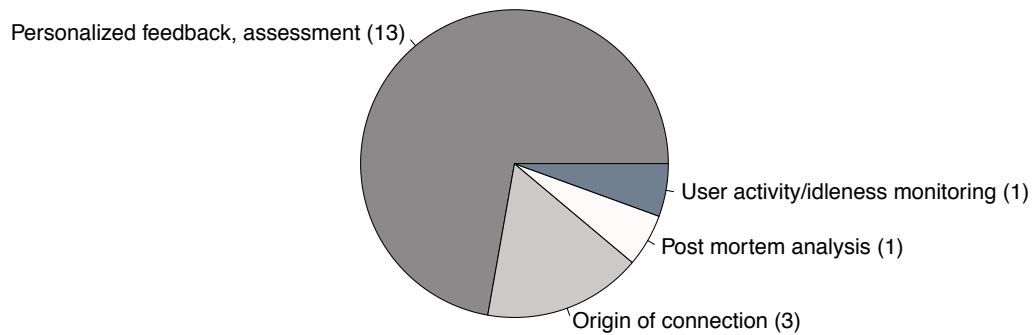
Q5: Has been useful to have activity tracking for your labs:



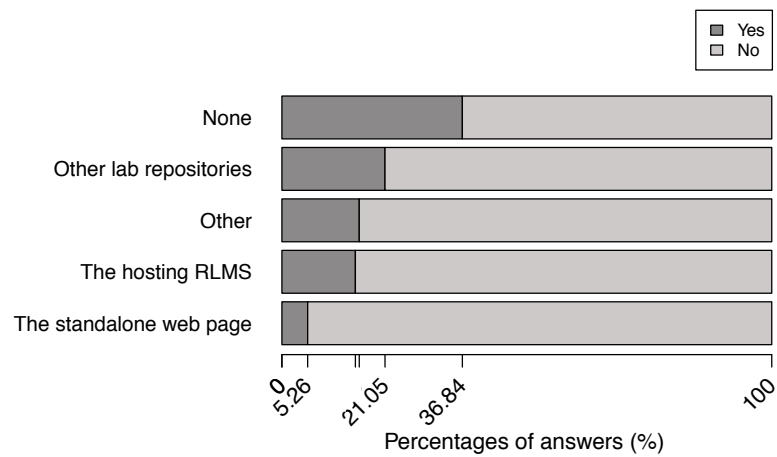
Q6: Why do you monitor your labs?



Q7: For which purposes do you monitor your labs?



Q8: Where do you publish your monitoring indicators?



Chapter 3. Activity Tracking Infrastructure for Embedded Cyber-Physical Labs

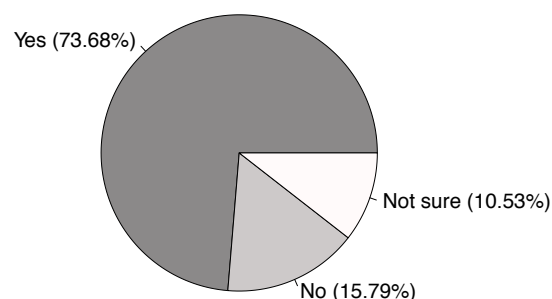
Summary

Only 1 of the 20 responders accounting for 5% of the responses didn't set up monitoring for their labs (Q1). The majority of lab owners answering this questionnaire have 5 labs or less (13 of the 20 responders), the rest had between 14 and 35 labs (Q2). When asked what do they track (Q3), in order, the answers were the connected users (78.95%), time of connection and disconnection (73.68%), values pushed to actuators and values read from sensors (63.16%), and 52.63% monitored for miscellaneous indicators: where the lab was accessed from (Learning Management Systems, direct access, or other), web browser used, IP address, geolocation, queuing duration, language used, internal events such as hardware interlocks, alarms, and others. In Q4, we wanted to know how frequently lab owners monitor for specific analytics: lab owners mostly monitor for lab statistics and failure/security audit (Q3a and Q4c), and less frequently for system modeling and load balancing (Q4b and Q4d). Q5 reveals that the responding lab owners tend to find monitoring the labs very useful (majority of ratings are between 5 and 7 points on 7-scale rating). The motivations found in the answers to Q6 for monitoring the labs are: because the technology is available (15.79%), because the hosting institution requires reporting (78.95%), and other miscellaneous reasons (47.37%). From Q7, we see that 13 lab owners monitor for purposes of "personalized feedback and assesment", 3 lab owners are interested in the origin of the connection to their labs, 1 "to make sure students are actually working with the lab" (user activity/idleness), and 1 lab owner is interested in a post portem analysis to understand what is making the equipment break down. 36.84% of the lab owners don't share their monitoring data, while others published them to a variety of channels: lab repositories (around 21%), same propotion for hosting RLMS (Remote Lab Management System), or a standalone lab page (5.26%), and 20% gave miscellaneous other answers (Q8). The lab owners who didn't publish their monitoring data said that they are not interested in sharing their data publicly, that they use them for their own internal reporting, or only published them in scientific papers.

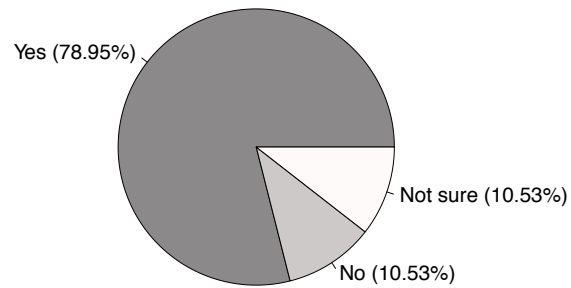
Part 2: Monitoring Dashboards

In this section, we want to know what lab owners want to use as advertisement for their labs on platforms. Three questions from this part are omitted as they are not related to our purpose in this work.

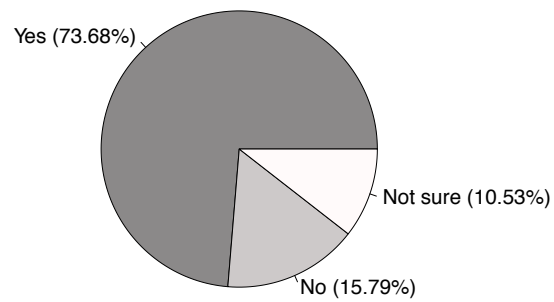
Q12: Are you interested in showing the total number of sessions?



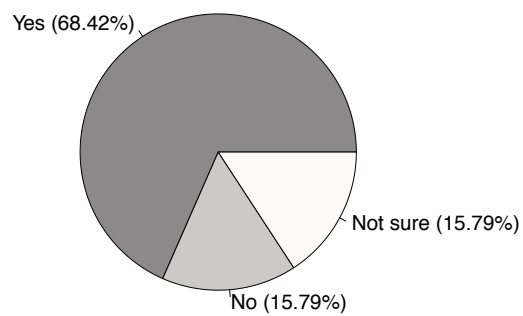
Q13: Are you interested in showing the number of unique users?



Q14: Are you interested in showing from which contexts the labs are being used (which MOOC, social media platform, LMS...)?

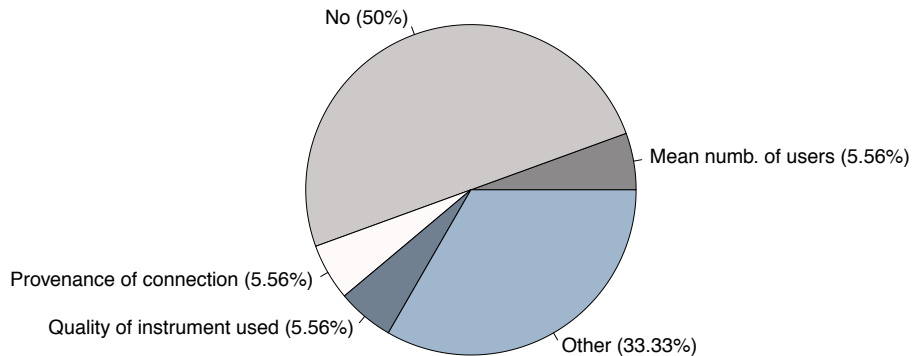


Q15: Are you interested in showing your up and down times on the lab repositories?

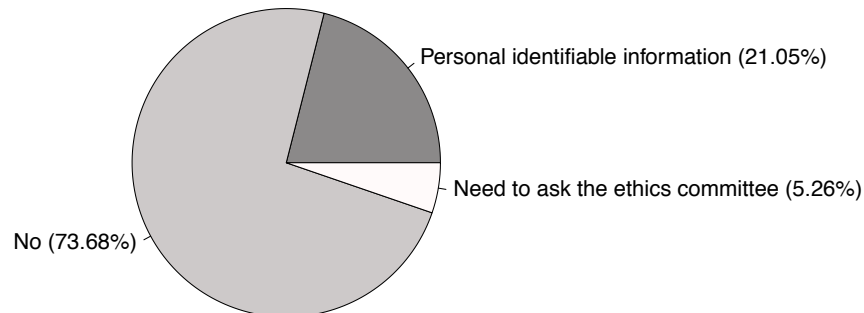


Chapter 3. Activity Tracking Infrastructure for Embedded Cyber-Physical Labs

Q16: Is there any other information you would want to publish to lab repositories regarding your labs?



Q17: Is there any information that you CANNOT publish because of regulations or other reason?

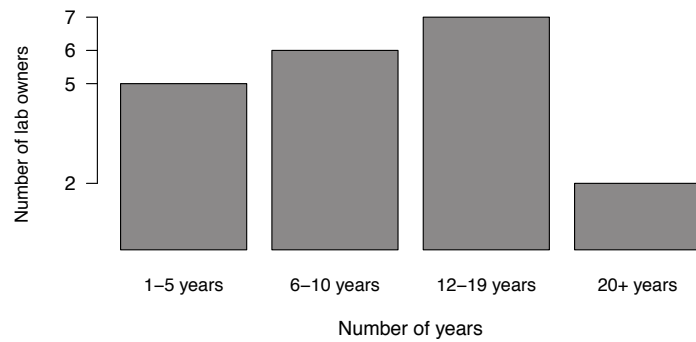


Summary

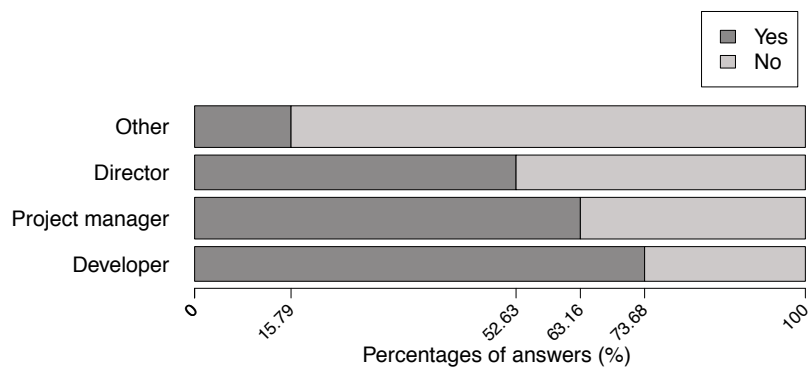
Around 74% of the responding lab owners are interested in publishing the total number of sessions for their labs, 15.79% said no, and 10.53% are not sure (Q12). The vast majority wants to show the number of unique users (78.95%), 10.53% are not sure, as well as 10.53% said no (Q13). 73.68% want to show from which web context the labs are accessed, 15.79% said no, and 10.53% are not sure (Q14). Regarding up and down times, 68.42% would show them, the remaining answers are equally divided between no and maybe (Q15). 50% of the respondents don't think there is more information to be published, the remaining 50% gave different suggestions (check Q16). For 21% of the lab owners, personal identifiable data cannot be published, 5.26% are not sure about what are data privacy restrictions, and around 74% don't have data publishing restrictions (Q17).

Part 3: Lab owner information

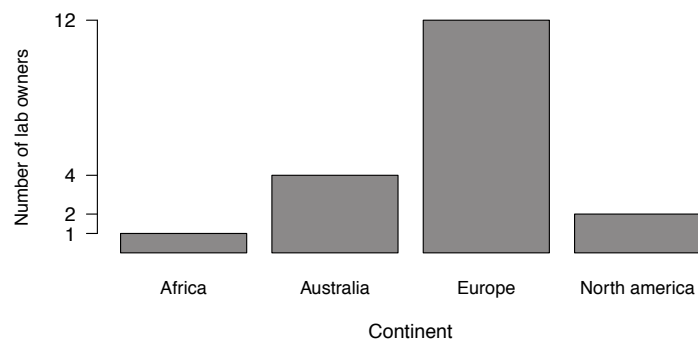
Q19: How long is your experience with remote labs?



Q20: What is your role in the development, deployment, and maintenance of remote labs?

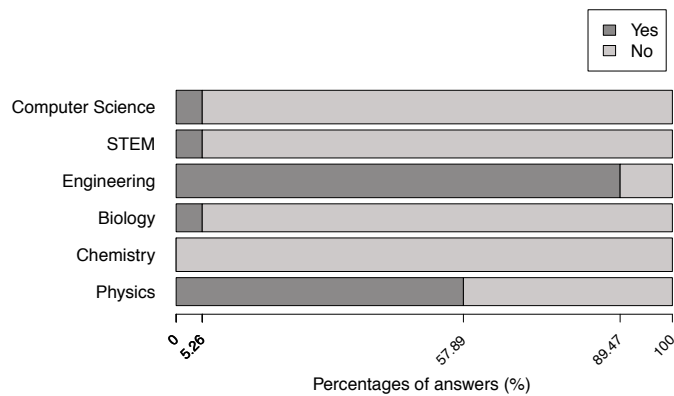


Q21: In which continent(s) are your labs located ?

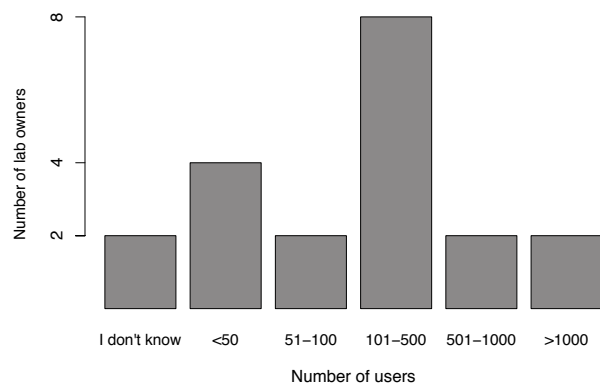


Chapter 3. Activity Tracking Infrastructure for Embedded Cyber-Physical Labs

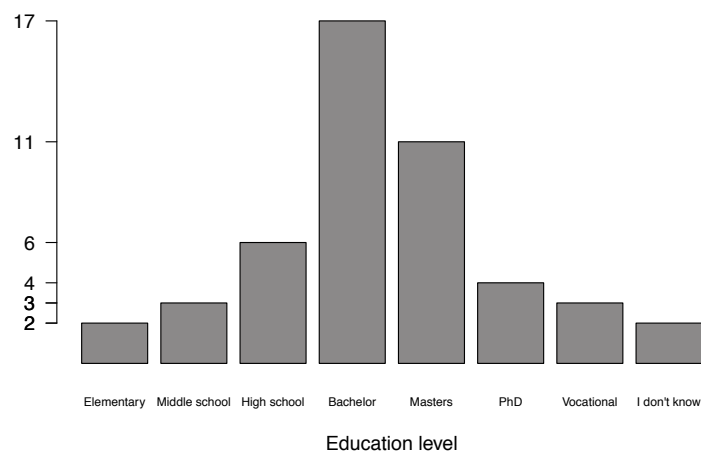
Q22: In which field are your labs?



Q23: How many users do you have each year?



Q24: What is the education level of your users?



Summary

The majority of the respondents have an experience working with cyber-physical labs for a duration of 6 to 20+ years (Q19). They have occupied a variety of roles in the cycle of developing and deploying these systems: developers (73.68%), project managers (63.16%), directors (52.63%), and 15.79% provided miscellaneous roles such as maintenance staff (Q20). 12 of the 20 responders have their labs located in Europe, 4 in Australia, 2 in North America and 1 in Africa (Q21). Around 90% of their labs support engineering applications, 58% have Physics labs, 5% computer science, STEM (5%) and Biology (5%) (Q22). No one is servicing Chemistry labs. 2 lab owners don't know their yearly user base, 4 service less than 50 users, 8 of the responding lab owners have a user base of 100 to 500 users/year, 2 have a user base of up to 1000 users a year, and 2 service more than a 1000 a year (Q23). 2 lab owners have labs for elementary school, 3 for middle school and 5 for high school. 17 lab owners support Bachelor-level CPLs, 11 service for Masters, 4 for PhD, 3 for vocational training and 2 haven't specified (Q24).

Discussion

The students answering the questionnaires detailed in Section 3.2.1 are adults who have the needed faculties to assess and judge their needs in online learning environments, and privacy issues. They have used the MOOC, hence they have been exposed to online learning situations, and can appreciate the difference between co-presence with others in the lab, versus being away connected through a computer to the lab. Even though the questionnaires were sent to the same pool and number of students, they were sent at different times of the academic year. The number of respondents was 33 for the first questionnaire which was sent during the semester, and 8 to the privacy questionnaire which was sent after the end of the semester. This explains the low response rate.

The 20 lab owners who responded to the questionnaire detailed in Section 3.2.2 have a variety of backgrounds, experience and responsibilities in the cycle of developing and deploying cyber-physical labs. They are servicing labs from a range of disciplines and are located in different parts of the world. Hence, we can say that they fairly represent the community of lab owners and their answers can be taken into consideration for the design of the activity tracking infrastructure.

In summary, we can conclude from the questionnaires sent to students and lab owners, that the two parties have different interests from the data generated of the use of CPLs. **Students** are interested in the interaction data in order to understand the dynamics of others using the labs, in sequencing this type of data and the corresponding experimental measurements in order to reflect and adjust their strategies— the same way they would do it when they are physically in the lab room with classmates. Their privacy issues are mainly regarding anonymity, data sharing and control. **Lab owners** are interested in the interaction for purposes of better maintenance, reporting and advertising. They comprehend privacy issues related to data collection and sharing.

In the next section, we analyze in detail the responses to the 3 presented questionnaires, and formulate the corresponding design requirements of the to-propose activity tracking infrastructure for CPLs.

3.3 Infrastructure Requirements

From the answers to the first questionnaire in Section 3.2.1, the students show us that positioning themselves relatively to others in the class plays an important role in detecting their correctness of procedures and results (Q1, Q6, Q7, Q8 and Q9) when experimenting away from classmates. Additionally, they give weight to checking with others to debug rather than resorting to lab instructor or assistance (Q4). In the second part of the questionnaire, they express their interest in being aware of the dynamics of the virtual class they are members of, and believe that the activity dashboards would be useful to emulate an experience similar to the one they have in class (Q10, Q11 and Q12). From the first two sections of the questionnaire, we can conclude that **students want awareness of the dynamics in a classroom** to aid them in reflecting upon their progress and adjusting their procedures. In the last part of the questionnaire, we understand that students follow a reasoning for sharing with, and asking for data from their classmates. An important portion of the students uses both indicators to debug their procedures. The students show a willingness to share their data (sequence and results) with others if needed, and they confirm that they would ask for that data if they needed it. **They also see a value in associating the experimental results with the sequences.** But they had a take on the need to be able to regulate the sharing strategies for either privacy or merit grounds (Q18, Q19 and Q20). So we observe that **students want to ask for others' activity data and associated results, and are willing to share theirs, but they want to be able to control the spreadability of information.** From this two requirements can be defined:

Requirement 1: activity traces should carry enough information to let students be aware of the dynamics in a CPLe experimentation activity.

Requirement 2: experimental results should be linked to a sequence of actions to enable the students to debug their procedures

From the students' privacy questionnaire, we see that **personal identifiable information remains the main concern for students.** While they are okay sharing their sequence, it appears that the respondents to the second questionnaire are not in favor of linkability (Q1). And that is supported by their preference for anonymous sharing of the data in Q4 of that questionnaire. Their trust in data management relies on their confidence in the Swiss laws and EPFL's regulations (Q2 and Q3). It is worth mentioning that in their case, the lab owner and lab instructor are the same entity, the institution hosting the labs and their university are the same entity, hence **if the lab providers and hosts are third parties, students might not trust that much in the system.** **Students feel strongly about merit as a parameter to regulate how their data is shared with classmates:** if the person shares their data they deserve to see others', if they don't they don't.

Revisiting the linkability of results with the sequence and the identity, students tend to accept to share all of this information only with those who do the same. Even though the majority of the students are willing to share their data, they prefer to have an opt-out option. And they are ready to share their data if that is the only way to see the dashboards (Q5, Q6, Q7 and Q8). Following this analysis, we can formulate the following requirements:

Requirement 3: students should be able to opt-in or opt-out on activity tracking.

Requirement 4: students should be able to regulate their data sharing in a modality similar to the real one.

Requirement 5: students should be able to trust where their data is saved and how it is used.

From the first part of the lab owners' questionnaire, we see that they mainly monitor to have lab usage statistics and failure/security auditing, and the majority publishes these data to different channels: hosting RMLS, lab repositories and others . From this, we can conclude that **lab owners need monitoring for their labs and publishing their stats to repositories**. In the second part of the questionnaire, we wanted to know which indicators they would publish to repositories and which they would refrain from. The majority of the respondents want to publish data related to usage statistics including number of users, sessions, up and down times of their services; but they would not or are not allowed to share personal identifiable information. So we can say that **lab owners want to publish their monitoring data to other platforms, and they need a mechanism to control which indicators to show or hide**. Hence:

Requirement 6: activity traces should carry enough information to enable lab owners to understand their labs usage.

Requirement 7: lab owners should be able to publish their data to repositories and control which indicators are sent there.

3.4 Infrastructure Vocabulary

The data structure describing an interaction plays a principal role in deriving meaningful insights from it. del Blanco et al. [19] identify two types of data formats in learning contexts: *static data* in the form of a student profile, or *dynamic data* in the form of student-performed actions with associated outcomes. Student profiles carry information regarding the age, interests, gender, residency and other personal parameters, while dynamic data mostly is structured following an *actor-verb-object* format. In our work, we are interested in gathering interaction data which relays understanding of the dynamics of a student and the collective class, hence we are interested in the *dynamic data* paradigm. Based on the comparison between dynamic data formatting schemes in Section 3.1, we adopt the xAPI¹¹ specification, since it provides guidelines on how to capture

¹¹<http://experienceapi.com/>

Chapter 3. Activity Tracking Infrastructure for Embedded Cyber-Physical Labs

user interaction with an online environment in a consistent way, by formalizing the formats of those traces according to activity domains and associated vocabulary to promote consistency and data reuse.

In this section, we detail the xAPI specification before introducing our proposed extension to its vocabulary. The extension we propose responds to Requirements 1, 2 and 6 elicited in Section 3.3, which state that activity traces should carry enough information to allow the extraction of insights needed by students and lab owners.

3.4.1 The xAPI Specification Explained

The xAPI specification defines guidelines for capturing online learning experiences such as watching a video, taking a quiz and others. Those guidelines are two-part: first the *statement design* or *structure* of the activity trace in its most basic form *actor-verb-object*, and the *vocabulary* (the word choice) which is used to write the statements describing each component. We start by explaining the parts of an xAPI statement through examples, then we move to detailing the guidelines for *statement design* and *vocabulary*.

Parts of an xAPI Statement

The parts of an xAPI statement are the building blocks which will form the activity trace describing an experience, for example watching a video or submitting a quiz. The xAPI specification specifies 7 fields: the *actor*, *verb*, *object*, *context*, *results*, *extensions* and *attachments*. Only the first three fields are mandatory for a valid xAPI statement, and are called the “core parts” of the statement, the rest is provided to support information granularity as detailed below [72]:

- **Actor:** the actor is the person doing the experience. An actor is uniquely identified by their ‘*mbox*’, making it an obligatory field for the *Actor* part of the statement. An example *Actor* is shown below:

```
"actor": {  
  "name": "John Doe",  
  "mbox": "mailto:john.doe@example.com"  
}
```

Listing 3.1 – An example Actor as formatted in an xAPI statement

- **Verb:** the verb describes what has happened between the *actor* and the *object* in a statement. It should be part of a statement, and it is identified with a resolvable URI. TinCan already provides a list of ready to use verbs in their registry¹². This is an extracted example from the specification’s documentation for the verb ‘experienced’:

¹²<https://registry.tincanapi.com/#home/verbs>

```
"verb": {
  "id": "http://adlnet.gov/expapi/verbs/experienced",
  "display": {
    "en-US": "experienced"
  }
}
```

Listing 3.2 – An example Verb as formatted in an xAPI statement

- **Object:** it is the entity upon which the *verb* occurred. It is an *activity* such as video watching if we are describing the global interaction, a *person* if the *actor* mentions another student in a lesson, or an activity component such as the video if the video is paused. An *object* should be uniquely identified with a designated URI as in the example below for video watching being the object:

```
"object": {
  "id": "https://registry.tincanapi.com/#uri/activityType/79",
  "name": {
    "en-US": "video"
  },
  "description": {
    "en-US": "Represents video content of any kind."
  }
}
```

Listing 3.3 – An example Object as formatted in an xAPI statement

- **Context:** a non-mandatory field to add contextual information to a statement. There are no constraints on what this field holds, in the example below extracted from the xAPI documentation, they include the instructor of the course:

```
"context": {
  "instructor": {
    "name": "Irene Instructor",
    "mbox": "mailto:irene@example.com"
  }
}
```

Listing 3.4 – An example Context as formatted in an xAPI statement

- **Results:** is a measurement of the outcome of an activity. For example, if John Doe watched a video in a given course and he completed the course successfully with a 95% of the full grade, this could be expressed as follow:

```
"result": {
  "completion": true,
  "success": true,
  "score": {
    "scaled": .95
  }
}
```

Listing 3.5 – An example Result as formatted in an xAPI statement

Chapter 3. Activity Tracking Infrastructure for Embedded Cyber-Physical Labs

- **Extensions:** this field can be part of the *activity*, *context* or *result* fields. It is provided to allow for special use by an application or as a convention by a community. This field supports the xAPI acclaimed statement design flexibility, not to limit the granularity a statement holds. In the following example, the *extensions* to the *results* informs us that the average grade on the test is 70 points.

```
"result": {
  "completion": true,
  "success": true,
  "extensions": {
    "http://example.com/course/averageGrade": 0.70
  }
}
```

Listing 3.6 – An example Extensions as formatted in an xAPI statement

- **Attachments:** this field serves to save files associated with an undergoing experience, it could be a certificate of completion as in the example below from the xAPI specification, or in our case, the experimental results the students would save.

```
{...
"attachments": [{
  "contentType": "application/pdf",
  "usageType": "http://id.tincanapi.com/attachment/certificate-of-completion",
  "display": {
    "en-US": "Completion of Experience API 101"
  },
  "description": {
    "en-US": "Certificate provided as proof of completion of Experience API 101 course."
  },
  "length": 63878,
  "sha2": "c2a36cbc4db66444d05e134b85a89681f65263cacd93eb4a544f0bef058a5649"
}]
}
```

Listing 3.7 – An example Attachments as formatted in an xAPI statement

Recipes and Profiles

Recipes and *Profiles* are meant to control the validity of statements' vocabulary and structure respectively. A *Recipe* [71] is a standard vocabulary to be used in describing an experience, i.e. which xAPI statements are used to track a particular type of experience. The main reason for having recipes is to avoid the use of different vocabularies for tracking the same kinds of experiences, which would fragilize the consistency of the xAPI specification and promote redundancy. This would also result in different applications recording the same kinds of experience in different ways, which will lead to the need of constructing custom reports for each application, violating the main goals of xAPI: interoperability of the specification. The complete list of

xAPI-ready-to-use recipes can be found on their webpage¹³. While the Recipes are the set of statements which describe an experience, *Profiles* define the expected values and the possible combinations between *verbs*, *results*, and *activities* for a given experience. For example, if the verb is *answered* in a tracking a quiz experience, the *result* field should hold a value for the score for that answer and a *success* field.

There isn't any *Recipe* for experimentation with a CPL as an online learning experience, and since we have two stakeholders in need of different answers from activity tracking, we define the Remote Experimentation Profile¹⁴ with two Recipes corresponding to the lab owners and the students, as detailed in the next.

3.4.2 Extending the xAPI Profiles and Recipes for CPL Experiences

The *Profile* we propose comprises two *Recipes*, which define the structure and terms used in the statements intended to capture the experience of using CPLs, when students and lab owners derive value from the data respectively. The proposed CPL-xAPI Profile has three sections: the *Basics*, *Activities* and *Verbs* detailed hereafter.

Basics

The *Basics* define the meaning of the fields which could vary in the statements for the CPL-xAPI Recipes:

- All statements include the *recipe ID* which identifies whether the statement is captured to serve the needs of the lab owners or the students.
- The *actor* is the person being observed: the student in both recipes.
- The *object* is either the CPL experimentation as the global experience, or a UI element of the CPL web app as seen by the students (e.g. slider).
- The *object* is either the CPL experimentation as the global experience, or a specific sensor/actuator as seen by the lab owners.

Activities

Activities describe the types of *activities* (as explained in Section 3.4.1), the recipes capture as perceived by the students and lab owners. We make the distinction on the level of granularity an *activity* can hold, hence defining *root* and *item* activities as follows:

¹³<https://experienceapi.com/recipes/>

¹⁴'Remote Experimentation' is officially used on the request for the extension request to ADL, because at that time the nomenclature adopted in this work was not CPL yet. Remote Lab Experimentation and CPL experimentation can be interchangeably used.

Chapter 3. Activity Tracking Infrastructure for Embedded Cyber-Physical Labs

- Root activity:
 - For lab owners: the *root* activity identifies the CPL equipment as a whole. These activities affect the whole CPL setup, such as a user connects and disconnects.
 - For students: the *root* activity identifies the CPL web app as a whole. These activities relate to an activity happening on the high level of the CPL web app such as the student opens and closes it.
- Item activity:
 - For lab owners: the *item* activity is exerted on one component of the lab setup, such as a sensor or an actuator.
 - For students: the *item* activity related to action on one element of the CPL, such as moving a slider.

In our case, the only *root activity* which will be used as an *object* in the statements where adequate is the experimentation with a CPL. Given that there is no formalization for this online learning activity in the xAPI statement, we formally proposed the addition of the *Activity* to the registry, the proposal was accepted, and can be found on the official xAPI registry of Activities ¹⁵.

Verbs

The xAPI registry contains a list of verbs with their definitions and their resolvable URIs (the unique identifies or xAPI verbs). The purpose in this work is not to re-invent the wheel, hence we make a selection from an existing list of verbs which correspond to our needs. However, one verb remains missing to complete the necessary verbs for our extension, so we propose to add a verb (*configured*). In the following we specify the usage of verbs in the respective contexts of lab owners and students:

- For the students:
 - *started*: indicates that the *actor* has started the experience. For instance, when the students connects to the CPL for the first time.
 - *resumed*: indicates that the *actor* is coming back to a suspended attempt on an *activity*. Should only be used on an *activity* that previously recorded a *suspended* statement.
 - *terminated*: indicates that the *actor* has terminated or exited the *activity*. For example, when signing out.
 - ***configured***: we add this verb to the registry. It indicates that the *actor* sent parameters to configure the experiment. When *configured* is used, it is expected to have an *extensions* field which records to value of the sent parameter(s).
 - *suspended*: used to describe the action of pausing an *activity* with the intention of returning (resuming).

¹⁵<https://goo.gl/2bFVyz>

- *read*: indicated that the *actor* asked for data from a sensor to be read. In this case, an *extensions* field is expected, which holds the value of the pulled value.
 - *saved*: indicated that the *actor* expressed interest in some result of the *activity* to be saved. For CPL experiences, the verb *saved* is used when the student attempts to save their experimental results. When using this verb in a statement, the *attachment* field should be present and it should hold the metadata and data to be saved.
- For the lab owners:
 - *started*: indicates that the *actor* has started the experience. In this case, it is equivalent to the user connecting to the setup.
 - *terminated*: indicates that the *actor* has terminated or exited the *activity*. For example, when disconnecting from the setup.
 - *used*: indicated that the *actor* sent commands to the actuators (the object) or wanted to read from a sensor.

3.5 Infrastructure Model

In this section, we satisfy requirements 3, 4, 5, and 7 previously detailed in Section 3.3. These requirements touch on the regulation of data sharing for both students and lab owners. We aim to satisfy their respective needs on an architectural level. Hence, before presenting our architectural model of the activity tracking infrastructure, we analyze the *activity data sources* and their role in implementing privacy mechanisms.

Analysis of Activity Data Sources

First, tracking the activity in the CPL experience can happen at two levels, which we refer to as the *data sources*: either at the client app side (the CPL web app), or at the server side (the LaaS layer or the server/lab owner infrastructure). Second, referring to the integration layers introduced in Section 1.5.2, Figure 1.14, user identification is not passed through the layers to the LaaS layer, and the lab owner does not have any personal identifiable information regarding the learner, making data collection from their side possible without any privacy concerns. In our architectural model, we define the implementation of activity tracking for students on the CPL web app side, and for lab owners on the LaaS side.

Supporting the Needs of Students

To support Requirements 3, 4 and 5 elicited in Section 3.2, which state that students should be able to control data tracking, sharing and consumption; we devise the following data flow plan: after authentication with the learning platform, the students get access to the CPL web app as part of their learning resources.

Chapter 3. Activity Tracking Infrastructure for Embedded Cyber-Physical Labs

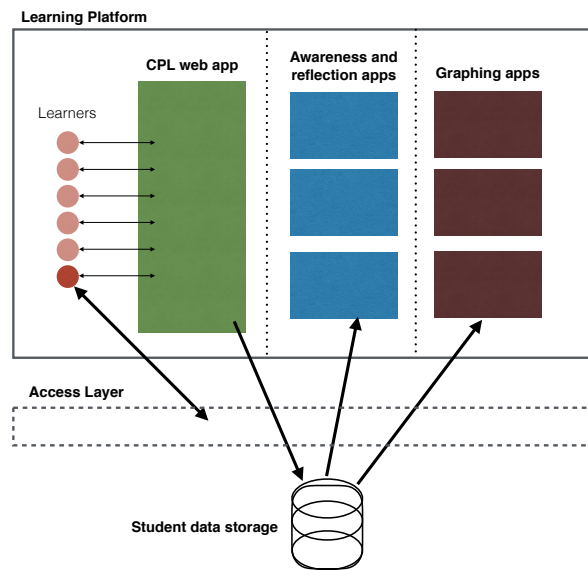


Figure 3.1 – Proposed activity tracking model for CPLs supporting the needs of students.

At this level, students should be able to choose whether to opt-in or opt-out on activity tracking. The privacy management panel could be implemented on the platform or the CPL web app levels. This is abstracted in Figure 3.1 as an *Access Layer* which manages the data flows between the source (CPL web app in this case) and the destination. In the case where the students opt-in for activity tracking, they are granted access to awareness and reflection apps which consume the data recorded. Then, the graphing apps retrieve data from the same data storage as the awareness and reflection apps. If the student opts-out on activity tracking, they will not be able to access the awareness and reflection apps, but they will still be able to use graphing apps. As the access to these apps should not be used for bargaining on activity tracking, otherwise, students are implicitly obliged to activate tracking to be able to complete their course work. When students opt-out on tracking, their experimental results are not saved in the data storage supporting activity tracking, but on the learning platform or through any other implemented data storage mechanism. The rationale behind conditioning the access to awareness and reflection apps is to incite students to contribute to the analytics data, helping in providing larger data for insights and hindering lurking behaviors. Figure 3.1 shows the model of the proposed architecture.

The sequence diagrams depicted in Figures 3.2, 3.3 and 3.4 show all the cases of data flows in this section of the architecture. Figure 3.2 depicts the data flow of students enabling data tracking: when the student connects to a learning platform to gain access to a CPL web app, if applicable the platform internally authenticates with the CPL web app. The student has to explicitly choose to be tracked by granting access to a data storage for saving the activities. Reciprocally, the data storage grants access to the CPL web app to push the activities. Once the data storage and the CPL web app can communicate, awareness and reflection apps in addition to the measurements graphing apps are granted access to the data storage, in order to retrieve relevant data (Figure 3.3).

In the event that the student opts-out on activity tracking (Figure 3.4), the data storage is not granted access to the activities exerted on the CPL web app, awareness and reflection apps are disabled. In this case, graphing apps retrieve experimentally generated data (the measurements) from an existing mechanism if supported.

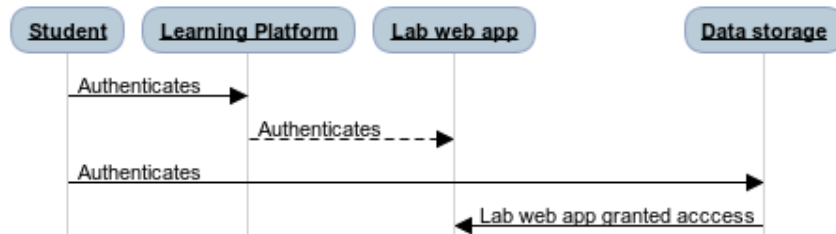


Figure 3.2 – Sequence diagram showing the student granting access to the data storage to save activity traces from the interaction with the CPL web app.

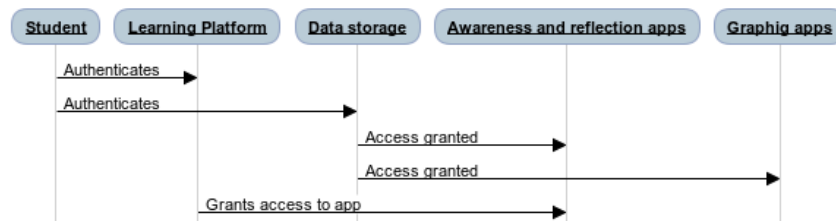


Figure 3.3 – Sequence diagram showing the data flow between the data storage, the awareness and reflection apps, and the graphing apps, when the student has granted access to activity tracking. In this case, all of the apps consume the data from the data storage.

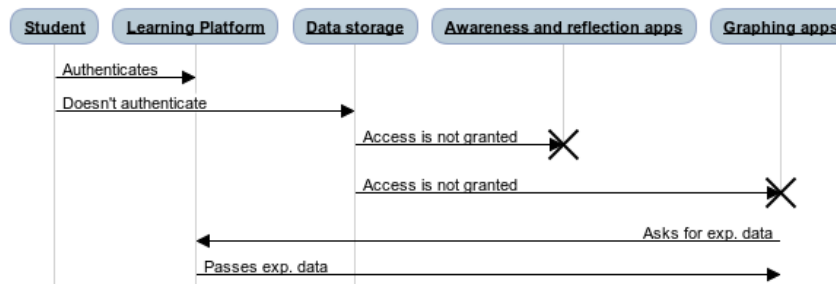


Figure 3.4 – Sequence diagram showing the data flow between the data storage, the awareness and reflection apps, and the graphing apps, when the student has not granted access to activity tracking. In this case, awareness and reflection apps are not functional. The graphing apps consume the data from a supported mechanism through the learning platform.

Supporting the Needs of Lab Owners

To support lab owners in their need to track and be able to pull their traces in lab repositories (Requirement 7 in Section 3.3), we propose the model depicted in Figure 3.5.

The Lab Owner Platform is composed of the Lab Server and the Lab Equipment. Once a user connects to the Learning Platform and the CPL Web App connects to the Lab Server, the lab owner detects the event and starts pushing the activity traces s/he wants to save. Accesses to the sensor and actuator services triggers the saving of action traces to the Lab Owner Storage. As explained earlier in this section, the lab owner does not need to have access to the CPL web app level where the identity of the user is known, hence any privacy issues pertaining to students is mitigated when the adopted data source is the Lab Owner Platform, where users are tracked, independently from the user and from the hosting platform. To export the collected data to analytics apps, such as those found in a Lab Repository platform, the lab owner decides what to publish. This is abstracted through the Access Layer in Figure 3.5.

The sequence diagram shown in Figure 3.6 depicts how data flows between the different components of the model. On their side, the lab owner grants access to a data storage where activities captured on the lab owner platform are stored. If lab owners decide to publish their analytics, they should grant access to the lab usage analytics apps.

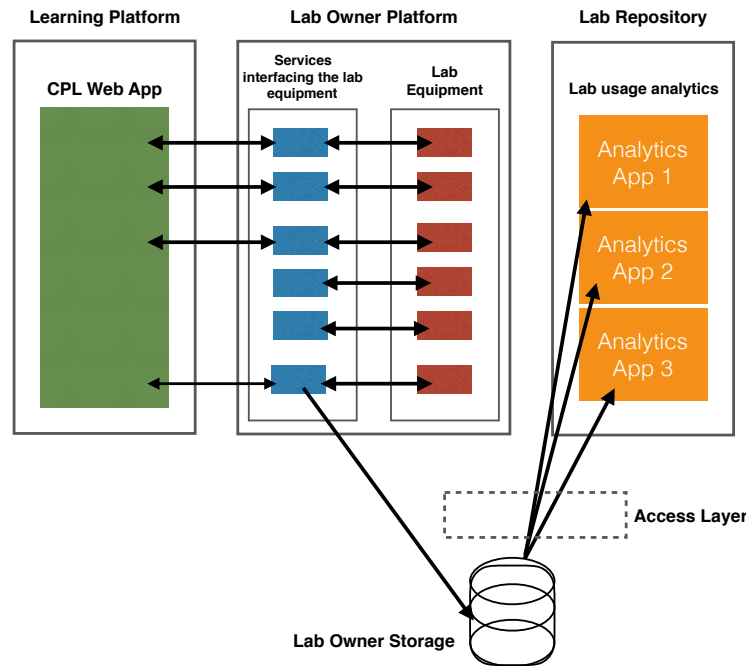


Figure 3.5 – Proposed activity tracking model for CPLs supporting lab owners

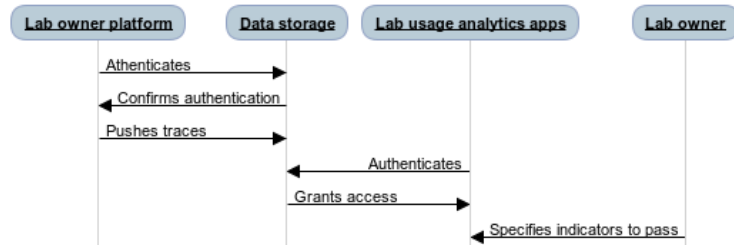


Figure 3.6 – Sequence diagram showing the data flow between the lab owner platform, the data storage, and the lab usage analytics apps, when the lab owner has activated activity tracking and granted access to the analytics apps.

Discussion

The models and the sequence diagrams used to explain the proposed infrastructure architectural model generalize design concepts, to be implemented in order to satisfy both the needs of students and lab owners. There is no indication as to where the location of the data storage is, for example, or which authentication mechanism is adopted. The proposed model shows how to decentralize the control of the user experience from the embedding platforms. The specifics of the implementations depend on the service providers (learning platforms and lab owners), and the available technologies. Example implementations of the proposed model are in the next section.

3.6 Implementation Examples

In this section, we present two implementation examples of the proposed activity tracking infrastructure. We first present the implementation of activity tracking on the user client-side with the Control Systems Lab MOOC, and then on the lab server side with the Mach-Zehnder Interferometer example.

3.6.1 Tracking Students' Activity on the Servo Motor - A User App Implementation

We consider the case of the electric servo drive CPL (detailed in Section 1.6.2), embedded in the Control Systems Lab MOOC, to demonstrate how the proposed architectural model in the previous section can be implemented on the CPL web app side, supporting the needs of students.

Chapter 3. Activity Tracking Infrastructure for Embedded Cyber-Physical Labs

The Data Management Panel

First, to make students aware of the possibility of being tracked, we make an announcement on the MOOC's main page. To inform them about where their data is stored and used, we added a tab for activity tracking configuration as seen in Figure 3.7, supporting Requirements 3, 4 and 5. The added tab leads to a simple activity tracking configuration panel shown in Figure 3.8. The panel provides the students with a detailed explanation of what will be tracked and where data is saved, in addition to information on how the data will be used if they activate tracking. This allows students to make an informed decision on whether they want to be tracked or not.

If they agree on activity tracking, the main page redirects to the page screenshot in Figure 3.8. They then gain access to the dashboard through this page.

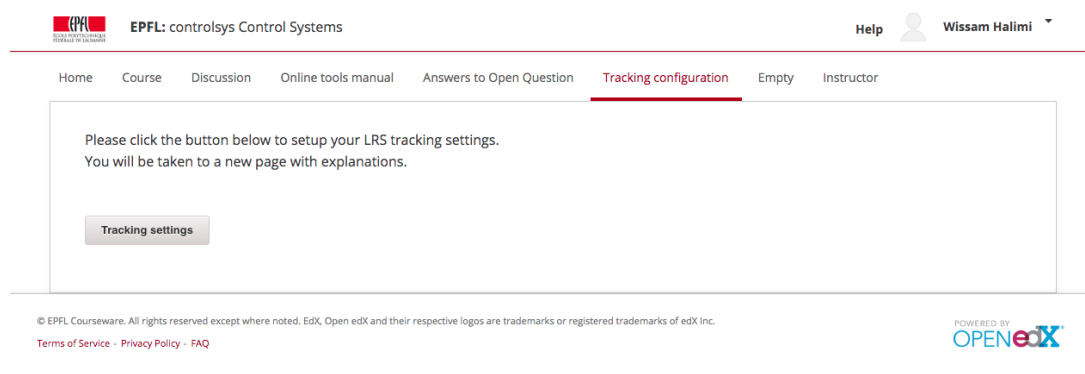


Figure 3.7 – The added tab which leads the learners to the activity tracking configuration page

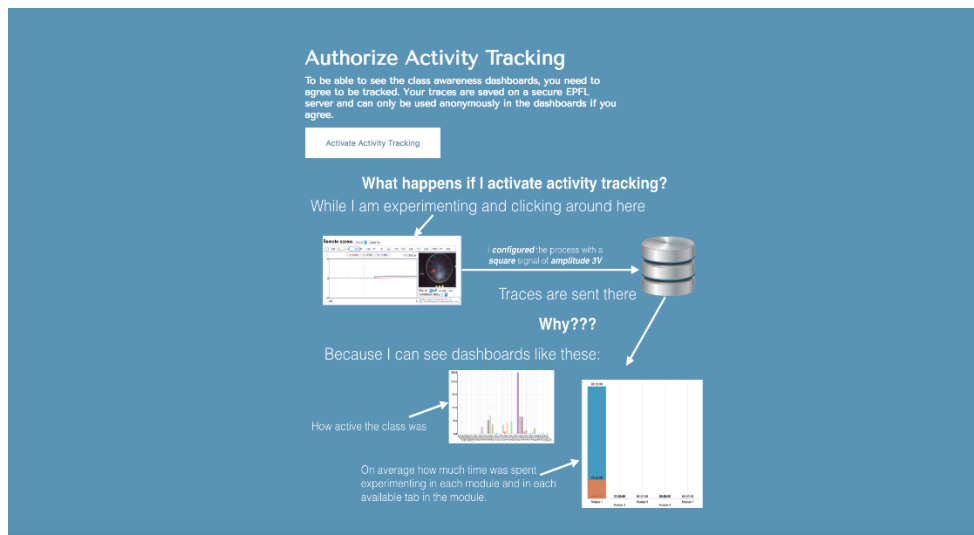


Figure 3.8 – Activity tracking configuration panel, the students are provided with information regarding the tracking mechanism, where their data is and how it will be used if they activate the tracking.

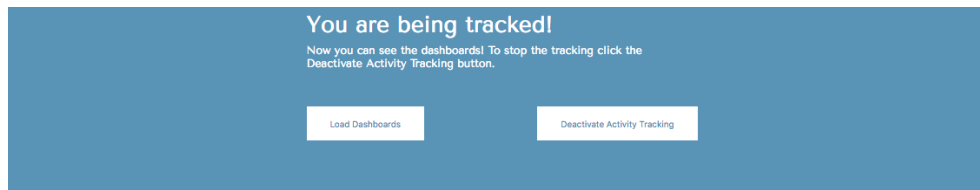


Figure 3.9 – If students activate tracking, they get access to the dashboard through the panel, and they can deactivate tracking at any time.

The Dashboard

Students taking the MOOC for the year 2017–2018 were provided with a dashboard comprising 6 graphs. A zoomed out overview of the dashboard is shown in Figure 3.10, with a Deactivate Tracking button on the top right of the page. The dashboard has 2 types of activity graphs: those which are representative of the collective activity of all those taking the course and using the CPL; and those which show the personal activity of the student. Each graph has a short description, which explains to students how to interpret the visualizations. In Figures 3.11, 3.12, and 3.13 we zoom in to show the details of 3 of the provided graphs.

Figure 3.11 shows one example graph, which provides a summary of how active was the access to the CPLs by all the users. In Figure 3.12, we show the equivalent dashboard, with the activity data of the logged user filtered and used. Figure 3.13 shows a detailed plot of how much time the class has spent experimenting in the modules of the course, grouped by experiments. Similarly to the previous examples, a version of this graph for the individual time spent experimenting for the logged user is provided. Such dashboards allow students to compare their activity to the collective activity of the class, and evaluate their contribution to the total. The xAPI statements used in this example according the proposed extension in Section 3.4 are listed in Appendix A, Section A.2.

The presented implementation was tested with the students on the Control Lab MOOC of the academic year 2017–2018. Initially, the aim was to evaluate students’ take on the proposed privacy management framework , and study the effect of having awareness and reflection tools (the dashboard) on their online experimental behavior. Due to lack of data, it was not possible to conduct such a study. In Section 3.7, we evaluate the implementation in order to understand why we were not able to collect enough data (i.e. why not enough users used the implemented infrastructure).

Chapter 3. Activity Tracking Infrastructure for Embedded Cyber-Physical Labs



Figure 3.10 – Zoomed out overview of the dashboard provided to MOOC students. There are 6 graphs. On the top right is the Deactivate Tracking button.

Lab work activity over the last 30 Days

The bar plot shows how active the class has been since the start of the course. The number of actions on the lab are plotted against the days. The more actions you and your classmates do while experimenting, the higher the number of actions. 21 people have contributed to this chart. To view in more detail the individual contribution to the activity, click on one of the bars and check the next bar plot.

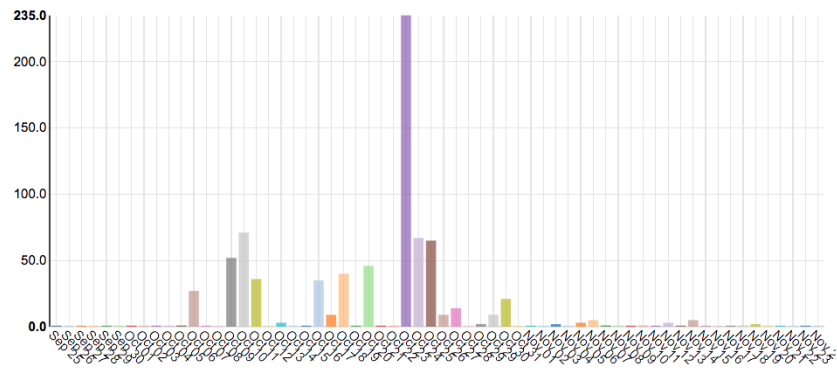


Figure 3.11 – Example collective graph, showing the number of actions exerted on the CPLs by the whole class, through a defined period of time.

My lab work activity.

Just like the first chart, this one shows the number of actions on the lab per day, BUT for you only. Each bar has the number of actions you did on the lab on the days you used it since the start of the course. How do you compare?

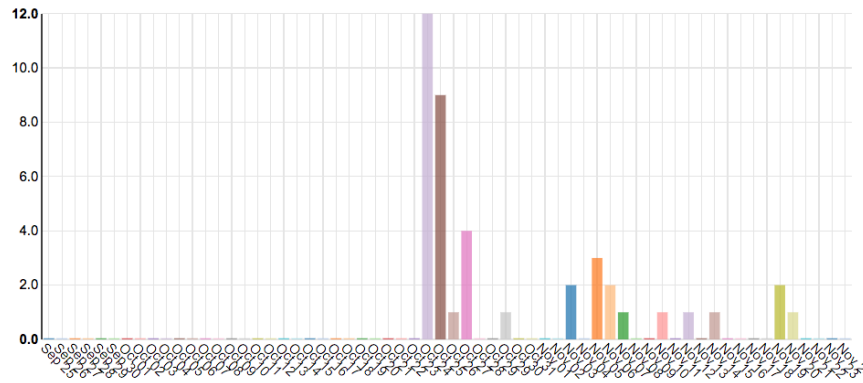


Figure 3.12 – Example individual graph, showing the number of actions exerted on the CPLs by the logged user, through the same defined period of time as in Figure 3.11.

Collective timespent experimenting in modules.

This chart shows how much time on average the class has spent experimenting in each module. Hover over the bars with your mouse to check the total time spent in each experiment if a module has multiple experimentation tabs. How do you compare?

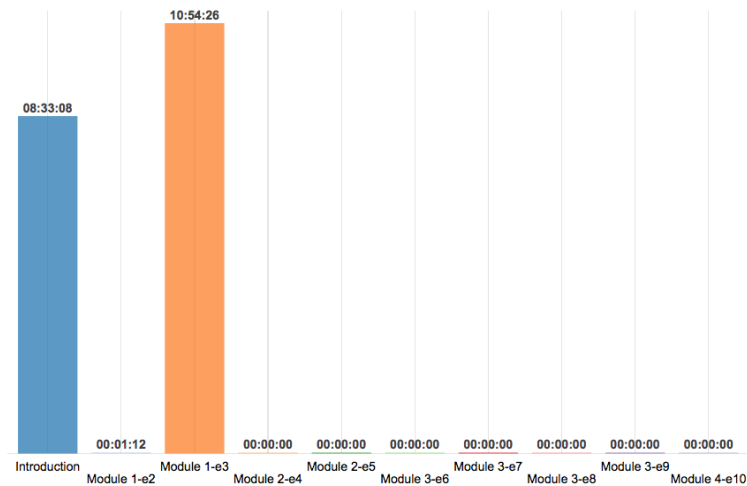


Figure 3.13 – Example collective dashboard, showing the total time spent by the whole class, experimenting in Module, grouped by experiments (Module 1-e2 denotes experiment experiment 2 in Module 1).

3.6.2 Tracking Students' Activity on the Mach-Zehnder Interferometer - A Lab Owner's Implementation

To show an example implementation of the proposed model on the lab owner platform, we revisit the Mach-Zehnder Interferometer CPL. Recall that the Mach-Zehnder Interferometer CPL is implemented according to the Smart Device specification, which corresponds in our case to assigning a single service to each sensor and actuator making up the CPL, bringing the total number of services to 6.

Setting Up the Communication between the Lab Owner Platform and the Data Storage

To start, we set up an xAPI-compatible database known as the LRS (Learning Record Store). There are several existing implementations of the LRS, we chose the most used at the time: the Learning Locker ¹⁶. An LRS only accepts and supports queries to xAPI compatible data, guaranteeing that when the activity tracks are logged, they are correctly formatted.

Then, the lab server side is modified to enable user tracking. A code to authenticate with the LRS is added: a new user id is randomly generated for each unique connection to the setup. The generated id is used to link the traces to connections, while the lab owner does not have the real identity of the user (to avoid privacy concerns). To trigger the of a new activity trace to be sent to the LRS, we embed a code to detect access and modifications to each service. Using the same randomly generated id, and the proposed vocabulary in Section 3.4, xAPI statements are built and pushed to the LRS. Appendix A, Section A.1 has the complete directory of all the xAPI statements used for this experiment.

The Dashboard App

We propose a web app which allows lab owners to pull their traces and use them to advertise their labs. Figure 3.14 shows the landing page of the app. The lab owner specifies from which LRS they want to pull their data, through the endpoint filled in the first field. To authenticate with the LRS, the lab owner should log in using their LRS credentials: a *username* and a *password* to be filled in the second and third fields respectively. The app has standard visualizations to showcase some indicators. The web app is in HTML, which can be easily embedded in Golabz for example as simply as in an iFrame.

Figures 3.15 and 3.16 show two example visualizations provided in the dashboard app. The first one (Figure 3.15) shows the number of unique connections per day to the CPL. The second one shows by order the most active users, and how many actions they exerted on the considered CPL.

¹⁶<https://learninglocker.net/>

Log In for LRS and Dashboards

<http://localhost:8080/data/xAPI/>

3cb95fe9d6f472137e35d778dcfcc1af45f78cb6

.....

☐ Remember me for this session

Sign in

Load Dashboards

Figure 3.14 – The landing page of the proposed dashboard app. Lab owners should use their LRS accounts to retrieve the data.

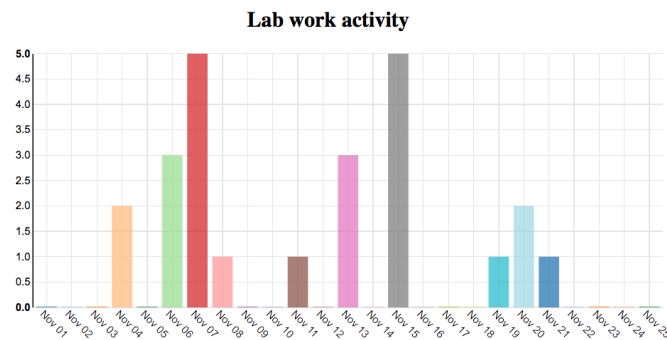


Figure 3.15 – Example of a lab owner visualization in the proposed dashboard app. Lab owners using this app can see the activity on their CPL by day. Each bar represents the number of actions exerted on the CPL.

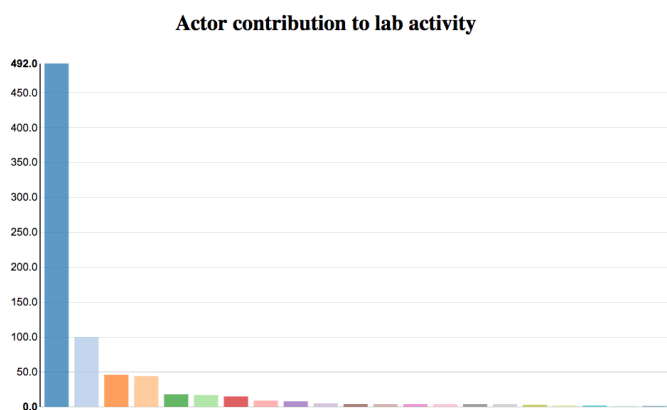


Figure 3.16 – Example of a lab owner visualization in the proposed dashboard app. Lab owners using this app can see how many actions users did on the CPL. Each bar is unique to a connection/user id.

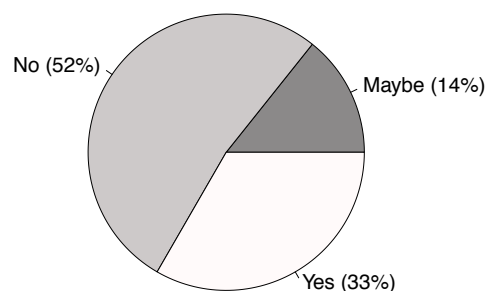
3.7 Evaluation of the Infrastructure with Students

The implementation example presented in Section 3.6.1 was available for use to the students of the Control Systems Lab MOOC for the academic year 2017-2018. Following positive responses of the students of the previous academic year regarding the usefulness of such an infrastructure, we were expecting for students of the current year to activate tracking. The prospectively collected data was to be used in studies, in order observe how students activating tracking and having the dashboards behave differently in their experimentation behavior, compared to the weeks where they didn't have the dashboards, and compared to the students who didn't activate tracking. But, after a month of deploying the implemented infrastructure and by the time of writing, not enough students had activated the tracking and only for a short period of time during the lab sessions.

Given that previous year's questionnaires were very positive about the potential of the activity tracking, and in light of the very low adoption rate of the tracking infrastructure by students, we evaluate the causes for such an outcome through a questionnaire. Were they not interested or are there other reasons? The questionnaire has two types of questions: first, questions repeated from previous year's questionnaires to indirectly test for the possibility that the students of the current year do not have the same interests in data, and reservations regarding privacy and data sharing as those of previous year. And second, direct questions to understand why they didn't activate the tracking, including system usability issues. The questionnaire was sent to 245 students enrolled in the courses using the MOOC for lab work. 21 students responded to the questionnaire (response rate of 8.57%). Hereafter we detail the questions and responses to the questionnaire.

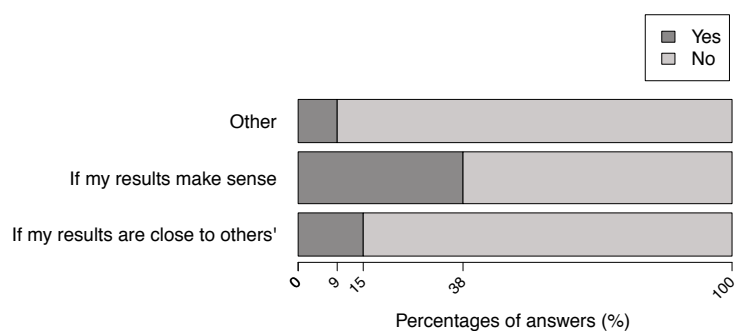
Part 1: Students' questionnaire

Q1: Are you interested in the progress of your classmates in the lab session?

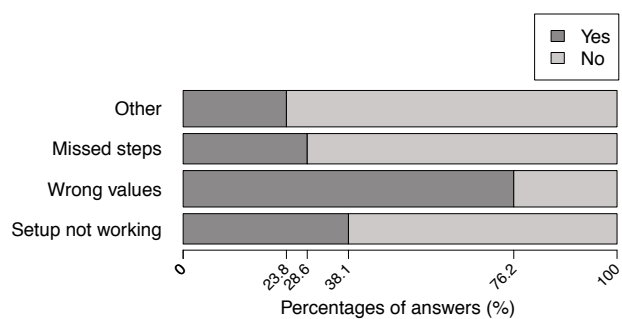


3.7. Evaluation of the Infrastructure with Students

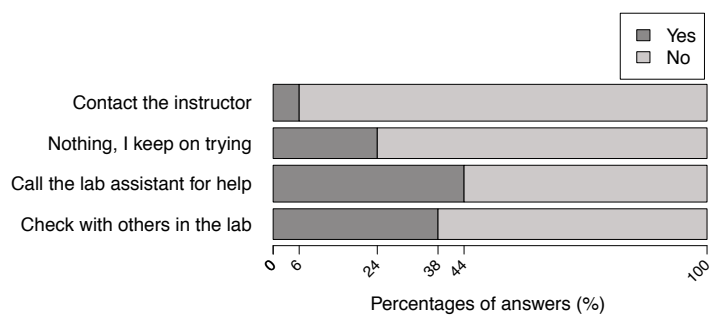
Q2: How do you know that you are on the right track in your experiments?



Q3: What kind of problems do you face when doing an experiment in the lab?



Q4: If you are in the lab, and having trouble with your experiments, what do you do?

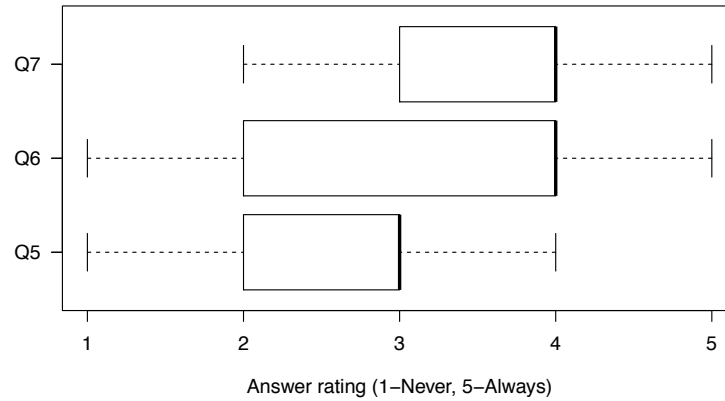


Chapter 3. Activity Tracking Infrastructure for Embedded Cyber-Physical Labs

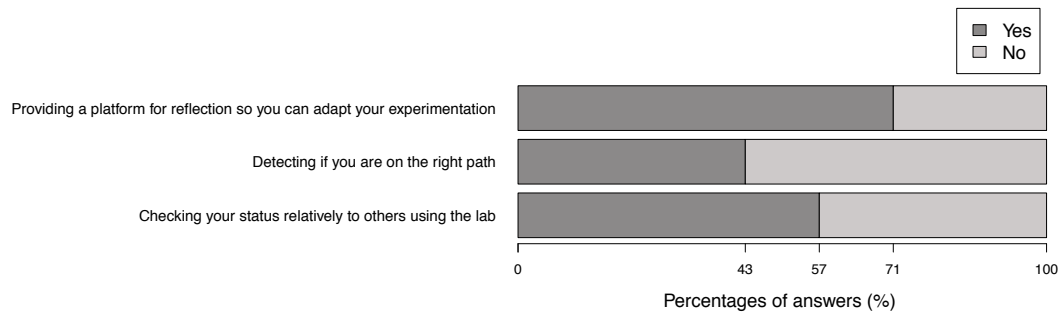
Q5: How often do you check with others their steps in the experimentation protocol, regardless of being in need for help?

Q6: How often do you check with others their experimentation results?

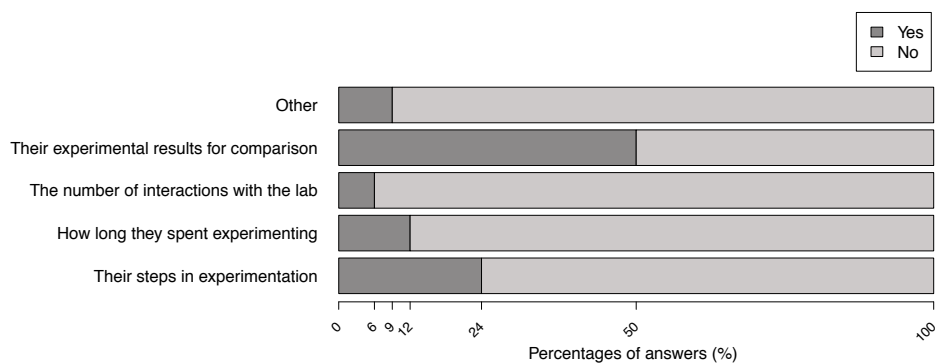
Q7: How useful those dashboards would be?



Q8: If you are doing your lab assignment online, away from the lab room, do you think such dashboards will help you in:

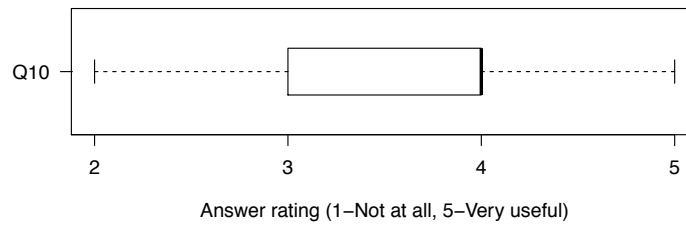


Q9: What would you like to know about others in the lab?

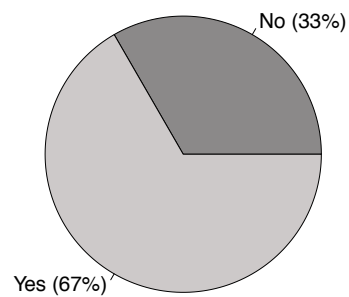


3.7. Evaluation of the Infrastructure with Students

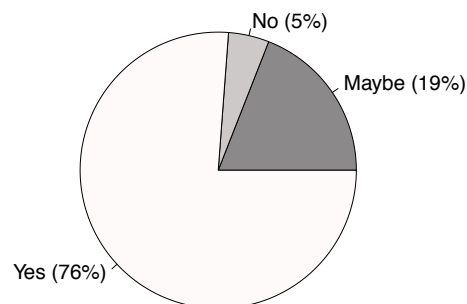
Q10: How useful the dashboards would be?



Q11: Are you willing to share with your classmates your sequence and data if they ask for help?



Q12: Would you ask a classmate for their sequence and data if you are running into trouble?



Chapter 3. Activity Tracking Infrastructure for Embedded Cyber-Physical Labs

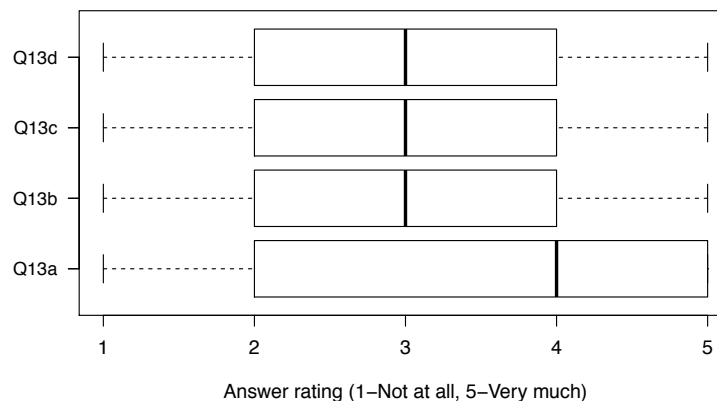
Q13: I am concerned about my privacy regarding:

Q13a : Disclosing my identity

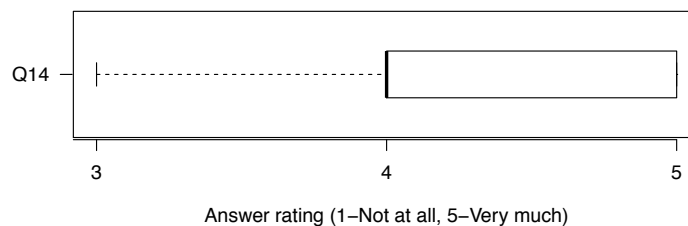
Q13b : Linking the sequence to me

Q13c : Linking the results to me

Q13d : Sharing my identity with class-outsiders



Q14: I trust that EPFL appropriately regulates my data privacy (i.e. doesn't share them with outsiders, will ask my permission for using the data, will anonymise the data.)



Summary and Discussion

Comparing Q1 of this questionnaire and Q1 of Part 1 of the student's questionnaire of previous year's in Section 3.2.1, we notice that current year's students are much less interested in the progress of others in the lab session (61% interested in the previous year vs. 33.3% this year). 38% and 15% of current year's students respectively refer to "if my results make sense" and "if my results are close to others'" to confirm their results, while previous year's students voted at 73.53% and 35.29% respectively (Q2 vs. Q2 of previous year's questionnaire). Q3 in this section and Q3 in Section 3.2.1 show that this year's students had less issues with defective setups (38.1% vs. 75%), relatively wrong values affect both pools of students the same (76.2% this years vs. 67% in the previous year). This year's students miss steps (28.6%) less than previous year's

3.7. Evaluation of the Infrastructure with Students

(45%). Their strategies to solve their inconveniences are proportionally different. Previous year's majority would check with others in the lab (83%), while only 38% of this year's do the same. 80% of previous year's students call for the lab assistant, while only 44% do this year. And last, contacting the instructor is done by 12% of this year's, vs. 6% of the previous year's (Q4 vs. Q4). On checking their steps and results, students of both years show similar trends (Q7 vs. Q5, Q6, Q7 and Q8 of previous year's). Notice that Q7 is formulated differently then Q5, Q6, Q7 and Q8 of previous year's, we wanted to avoid the details for our purposes; hence Q7 carries information we can deduct from the 4 questions we mentioned.

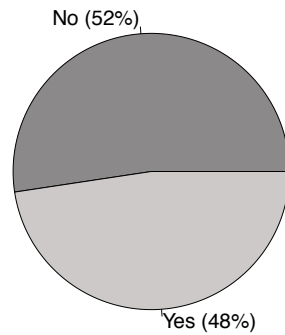
Their expectations from activity dashboards are comparable (Q8 vs. Q10 from part 2 in Section 3.2.1). Same proportions think dashboards provide a platform for reflection to adapt procedures (71% this year's vs. 73% previous year's). This year, 43% think dashboards would help in detecting if they are on the right track (73% previous year), 57% think they can help in checking their relative status to others (vs. 64% from the previous year). From Q9 we see that in the first place, students this year are interested in other's experimental results for comparison (50%), while last year 79% had that interest (Q11 from part 2 of previous year's questionnaire). 18% are interested in the number of interactions, 12% in how the long others spend experimenting, and 24% their experimentation steps, vs. respectively for previous year's: 18%, 32%, and 65%.

Both pools of students think comparably the same about the usefulness of the dashboards (Q19 vs. Q11 from previous year's), with both medians at 4 and the distributions of ratings tend to the left. More students from the previous year were willing to share their data (82% from Q18 of students' questionnaire of the previous year), while 67% are willing to share this year (Q11). From Q12 of this year's and Q19 of previous year's questionnaire, we see that same proportions of students would ask for help if in trouble (76%). Privacy concerns show similar trends Q13 and Q1 of the privacy questionnaire in Section 3.2.1. Student's rating for trusting EPFL are comparable, with both medians at 4 and the answers' distributions tend to higher ratings (Q14 and Q2 in Section 3.2.1)

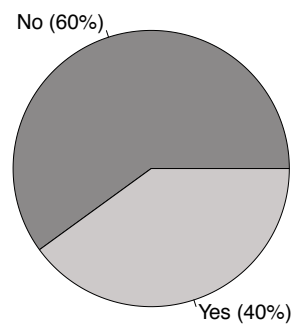
In summary, we notice that students this year are less interested in the dynamics and the intermediate actions leading to the end result of the experiments. This year's students are more interested in the results, and checking the dynamics seems to be only valuable if they need it once they realize that they are not on the right track. Additionally, previous year's students seem to be more proactive regarding debugging their procedures (checking with others vs. calling a lab assistant). The results from this part could be an explanation on low adoption rate of the activity tracking.

Part 2: Dashboard usability questionnaire

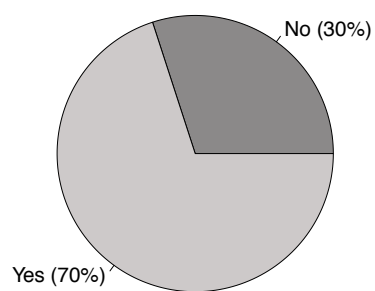
Q1: Did you activate tracking?



Q2: Were you aware that you had to activate tracking every time you connected to the lab?

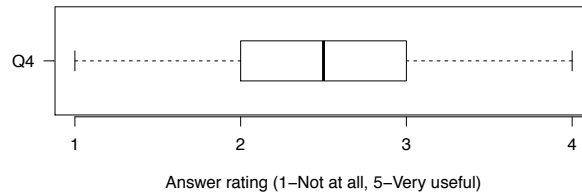


Q3: Were you aware that you can deactivate tracking anytime?



3.7. Evaluation of the Infrastructure with Students

Q4: The dashboards were useful:



Summary

Of the 21 respondents to this questionnaire, 11 activated the dashboards (Q1). Only 40% of those who activated the tracking at least once were aware that they should activate it each time they connected (Q2). As part of the privacy management mechanism, we automatically deactivated the tracking when a learning session is closed, but it seems that the students were not aware of it (only 30%). On the other hand, the majority of them were aware that they can deactivate at any time (70%, Q3). The usefulness of the provided dashboards ranked medium, with a median of 2.5 and the distribution of the ranking equally distributed between 1 and 2.5, 2.5 and 4 on a 5-scale rating (Q4).

Why students didn't activate activity tracking

Q1: I didn't activate because:

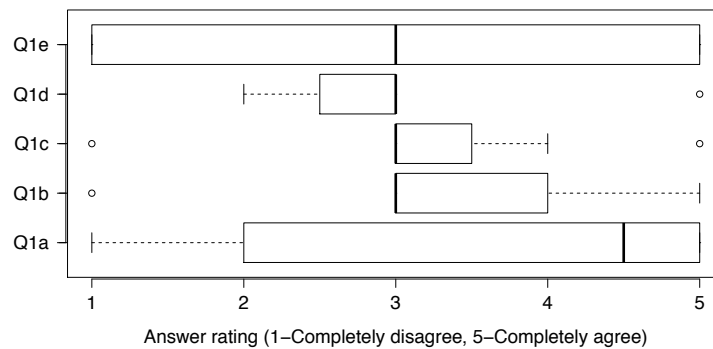
Q1a : I don't understand how it works

Q1b : Too many steps to get the the dashboards

Q1c : I don't think it's useful

Q1d : I think it is useful, but I am in contact with my classmate, the information is not new to me

Q1e : I am concerned regarding my privacy



Summary

From the answers of the 10 respondents who didn't activate the dashboards, we can say that the main reasons for them to opt out are related to usability issues (Q1a and Q1b). There is a tendency to think that it is not useful to have the dashboards (Q1c), but the fact that they are in proximity of their peers is not the reason (Q1d). They are equally divided on the “*I am concerned regarding my data privacy*” (Q1e), which would suggest that our privacy management modality proposal is not completely reassuring for them.

3.8 Discussion

Capturing, storing and consuming activity data generated from the use of embedded CPLs depends on a stack of components, of which: architecture, syntax and the privacy management mechanisms. The architecture determines at which level of the user interaction the data is captured (user web app or CPL server), and where it is stored (embedding platform level or elsewhere). The syntax defines the elements of the tracking trace (what information is kept), and in which format (proprietary, standardized or other). The privacy management mechanism dictates who is tracked (user consent or governmental/institutional laws), and guarantees or not the access of the data for consumption (which apps can consume it and for which purposes). How these three components are designed plays an important role in responding to the needs of the concerned stakeholders (students and lab owners).

To define the specifics of these three components for students and lab owners (**contributions 2 and 3**), we formulated **RQ2** and **RQ3** at the beginning of this chapter: **what are the experience data that students and lab owners are interested in?** And **what are students concerned about in their online privacy?** The aim is to not make assumptions about the needs of the stakeholders, and start from a user-oriented design. Through an elicitation of requirements, we found that students are interested in the progress of others using the CPLs, in keeping track of their measurements and in being able to control their privacy. Lab owners are interested in keeping track of the use of their infrastructures to better maintain them and advertise for them.

In light of the differing needs of both stakeholders and the privacy concerns of students, **RQ4** was formulated: **how to design an activity tracking infrastructure which responds to the needs of both?**. That, in the effort to respond to the needs of students and lab owners, while mitigating privacy concerns which could arise when trust between students and lab owners cannot be established on an institutional level (students and lab owners are not governed by the same data privacy regulations).

The proposed vocabulary in Section 3.4 extends the xAPI specification to track the activity of using CPLs in learning platforms (**contribution 4a**). The extension to the specification formalized the modality to track students when experimenting online, adding value and potential bigger and consistent data sets to learning analytics. The choice of the xAPI specification is based on the possibility of using the specification regardless of the embedding platform, a limitation of Caliper [11] discussed in Section 3.1.2. For the latter case, for traces to be usable in tools, they have to be in conformance with Sensor API, and the portability of the traces between tools and reuse of students' records in other platforms is not possible. Other general-purpose formats like ActivityStreams¹⁷ are re-purposed for educational uses, but there is no way to define a unified way for CPLs activities, as similarly as with xAPI. With the proposed extension, the interoperability and portability of students' generated data is possible. In [21], the authors detail compatibility tests runs on three major LRS providers: Learning Locker, Watershed LRS¹⁸, and WaxLRS¹⁹. They show how many modalities can be implemented to migrate the data to and from the different LRSes. This proves that if students can choose between LRSes from different providers to store their data, the operation of all the awareness and reflection applications will not be limited or hindered. In this case, a major consideration should be handled to allow students to re-use their data in platforms or tools than the originally provided: identity linked to their traces, and access to the storage system used by the CPL. The identity used by students to identify their traces should be consistent across the tools and LRSes, which is a technological limitation at this time, where LRS solutions are designed for the use of application providers but not users.

The proposed architectural model in Section 3.5 defines the levels at which activity is captured and data flows between CPL web app, lab owner infrastructure, data storage and data consuming apps (**contribution 4b**). Through the architecture of tracking CPL experimentation activity, we mitigated privacy concerned which could arise in situations where students and lab owners cannot establish grounds for trust. Given that students, through their use of the learning platform grant access to their identity, and through the platform access to CPL is granted, we specified that tracking CPL activity for the purposes of students should happen at the level of the CPL web app. We defined a mechanism to give students control over data saving and reuse, where access to awareness and reflection apps is given on the basis of merit (if a student contributes to the analytics data by activating tracking, the student gets access to the apps). For lab owners tracking, the assumption is that student identity is not passed to the lab owner infrastructure in the case of third-party providers. Lab owners track on their side the use of CPL random association

¹⁷<https://www.w3.org/TR/activitystreams-core/>

¹⁸<https://www.watershedlrs.com/>

¹⁹<http://www.saltbox.com/>

of connection to traces, enabling them to track activity on their side, choosing where to keep their records, and which platforms to publish their analytics on. Previous works for tracking students in online learning environments (except for MOOC platforms) are based on research purposes in education and pedagogical design [40, 76, 77] rather than user experience (which is our approach). Hence, the resulting architectures are either proprietary to the correspondingly designed platforms and cannot be extended ([40]), or limit the control of users to answer to research purposes ([76, 77]) as discussed in more detail in Section 3.1.

3.9 Conclusion

In this chapter, we elicited the needs and analyzed the concerns of students and lab owners from the interaction data with cyber-physical systems. We learned through the questionnaires (**contributions 2 and 3**) that students are interested in their progress relative to the class, that their main concern regarding data collection and retrieval is being able to control who can benefit from it on the basis of merit. Lab owners are interested in gathering data for purposes of equipment maintenance, system report and advertising. Taking into consideration the differing needs of students and lab owners, we proposed an activity-tracking infrastructure for the collection and retrieval of the interaction data with CPLs (**contribution 4**). The proposed architecture is accompanied by an extended xAPI vocabulary for activity traces formalizing xAPI statements for experimentation with CPLs. The proposed activity extension has been submitted to ADL and is now accepted, the *Profiles*, *Recipes*, and added verb are still under review. Later in the chapter, we provided two implementation examples with the electrical servo drive and the Mach-Zehnder interferometer. The electrical servo drive implementation was evaluated with students. Through the evaluation, we identified the reasons for the observed low adoption rate of the solution. It appeared that students who didn't utilize the implementation of the proposed infrastructure had problems with the usability of the implementation (Section 3.7).

4 Computational Analysis of Students' Access and Use of CPLs

The data generated from the interaction with the learning platform in digital education settings is used for a variety of purposes, ranging from personalization and recommendation to learning students' learning behaviors. In that regard, MOOCs present a mine of data which can be captured at different levels of the learning experience. On a high-level, times of logging in and out the platform are recorded, on a lower level sequencing through the resources in a lesson is tracked. Students are rather passive than active when using these resources. They are at the receiving end, either watching a video or reading lecture notes, except when they participate in forums and discussions. Evaluation methods such as graded assignments and quizzes record the answers, but don't provide a medium to understand how the students reached those answers, which is an important factor in inferring learner types, and getting more granular understanding of how students think. Even though the importance of lab work in scientific and engineering curricula has been long advocated as a factor contributing to positive academic performance in relative subjects and scientific thinking [26, 42], to date there are no standard criteria to evaluate the impact of lab work on academic performance, nor lab work has been much integrated in MOOCs.

The development of the CPLs integrated in the Control System Lab MOOC (presented in Section 1.2.2) was motivated by providing students with the possibility to experiment, unbound by the constraints of space and time. This is due to the potential of scaling the access to the 25 experimental setups, to the expected number of enrollments in the course for a semester, which is around 200 students. In this chapter, we consider the case of the Control System Lab MOOC of the fall 2016 semester, to investigate the following research questions:

RQ5: How are students accessing and using CPLs made available to them 24/7?

RQ6: How does students' experimental behavior impact their academic performance?

Through the collected data, we want to understand a number of exerted behaviors by students, which we group under two main banners: the **access to the CPLs** pertaining to the actual usage of the infrastructure and its success in servicing the prospect massive numbers of students, and

the **use of the CPLs** related to the experimental patterns they adopt in experimenting and their effect on academic performance.

This chapter is structured as follows: first we review existing work on the analysis and visualization of learners' data. Next, we detail the settings of the Control System Lab MOOC and how the Servo Drive CPL example introduced in Section 1.6.2, was integrated and scaled for the course [62]. We then analyze the access of the students to the CPLs during the academic year 2016-2017, and how it influenced their academic performance. Last we reflect on the obtained results and conclude.

4.1 Related Work

Since their emergence, MOOCs have been serving as rich data sources for researchers interested in understanding students' learning approaches and evaluation of academic success indicators. The data is used for both supervised and unsupervised machine learning approaches, for tasks such as student knowledge modeling [14, 57, 84] to fuel personalization mechanisms in MOOCs, such as ITS (Intelligent Tutoring Systems) [1] and recommender systems [53], or for studying the impact of learning approaches on academic performance [16, 28, 32, 75], or to simply discover latent learning patterns [66].

In [16], the authors detect whether online students of 4 different MOOCs are adhering to the designed learning path in the MOOC, and how their behavior impacts their success or failure in the MOOC— the designed learning path being the linear progression through a MOOC's tabs as organized by the instructor. They conclude that passing learners deviate from the designed learning path much less than non-passing learners.

When fitting a 4-state two-layer hidden markov model to the retrieved sequential student behavior data, the authors in [32] were able to make a distinction between two passive states (logging in the MOOC and doing nothing else, and browsing back and forth between the lectures and wiki), and two active states (actively browsing between lectures, wikis and submitting quizzes; and forum browsing for discussion and wikis). In their study, they find out that high-performing students have higher probability of being in the active states than the collective pool of learners, while their probabilities for being in the passive states are slightly less than those of the collective pool of learners. Low-performing learners had an increased probability of being in an active state related to actively browsing the contents of the MOOC (lectures, wikis and forums), which authors associate with 'help-seeking' behavior, a pattern which they speculate high-performing wouldn't seek, also in conformance with the conclusions of [45].

In [28], the authors explore how the learning approaches students adopt have an impact on their academic performance. They distinguish between three approaches: starting with video watching, starting with assignments and a mixed approach. They find no meaningful variation on the assignment grades, the number of submissions and time between submissions. These results lead

the authors to a conclusion in further accord with the conclusions of [32] and [45], that those who attempt the assignment before watching the videos with no difference in performance than others, already have the necessary knowledge and do not seek help. On the other hand, they observe that students which have a mixed-approach are score better than those who solely adopt the video-first or assignment-first approach.

The authors of [66] are solely interested in the discovery of latent learning patterns in a MOOC. Through data-driven and pattern-driven approaches they investigate learners' patterns during the assessment period of that MOOC. They find that 40% of the learners watch videos prior to attempting and submitting an assignment, and only 2% stick to not watching the videos at all throughout the course of the study. They also show that some learners permanently switch to a new approach than the originally adopted. They claim that detecting a change in the learning approach can be used for ITS for personalization purposes when students are facing difficulties.

So far, we have reviewed works which look at the collective behavior of sequencing a MOOC, with little or no attention to what students are doing in each step of their progression through their learning path. To the extent of our knowledge not many MOOCs offer tasks which are interesting to investigate. Most offered resources are passive to the student (video watching or lecture reading) except for evaluation tasks such as assignment and quizzes, where the outcome is evaluated rather than the problem solving approach. Few MOOCs offer interactive exploratory activities such as experimentation as in the case of the Control Systems Lab MOOC. The only works which we stumbled upon are by Fratomico et. al. [30, 31] and Venant et.al. [75].

The authors of [31] build Tempr, a tool which enables educators visualize learner's use of online labs by configuring the plotting parameters and grouping the students according to certain characteristics (for e.g. academic performance). Through a study conducted in the cited work with a simulation of an electric circuits lab, they find that when merging all tracked events, high-learners and low-learners appear to behave the same. Then, through selection of certain events, for example the use of a voltmeter and an ammeter, high-learners perform more active testing compared to low-learners with ammeters and voltmeters. Furthermore, checking the use of a voltmeter vs. the use of an ammeter shows that high-learners adapt better to the requirements of the exercises by switching over time to the use of the ammeter; while low-learners kept on using the ammeter as well.

In [75], the authors try to reveal relationships between learners' behavior during practical learning sessions and their academic performance. They perform sequence mining on the actions students do during coding sessions for an introductory course to shell commands. The authors categorize students' behavior according to 8 groups which they classify as confirmation, progression, success-then-reflexion, reflexion-then-success, fail-then-reflexion, trial-and-error, and withdrawal learning strategies. Through hypothesis testing with ANOVA (Analysis of Variance) they find significant results for the progression, success-then-reflexion, reflexion-then-success, and fail-then-reflexion strategies (those are the trends adopted by high-level learners); while they don't find any characteristic pattern for low-level learners. They also find a correlation between those

patterns and the grades students score on the considered exercise. Another result they obtain is that both groups of students homogeneously withdraw from an exercise, hence not identifying with the possible performance of a student.

4.2 The Control Systems Lab MOOC

4.2.1 Logistics

In the MOOC, students have lectures summaries, videos with instructions for experimentation to watch, quizzes to take, and remote access to the physical labs interleaved in tabs. Each lesson or module can comprise one or several tabs with remote access to the CPLs. Typically, the student opens the MOOC and goes through a sequence of tabs as shown in Figure 4.1, each tab with material to study. The complete course is composed of 8 modules. The considered Control Systems Lab MOOC is designed to be used in a flipped-classroom modality: the students are expected to get prepared for the exercises session by watching the tutorial videos; and then do the experiments during the allocated time at EPFL, during which teaching assistants are present. The MOOC is continuously available 24/7 regardless of the pre-scheduled lab sessions.

The MOOC is supposed to be used by students of two EPFL courses: Automatic Control and Control Systems. The total number of students taking both courses is 209 for the considered semester. Table 4.1 shows the distribution of the MOOC student pool. We notice that not all students taking the courses are enrolled in the MOOC (77.99% enrollment). This could be explained by the fact that students are allowed to work in groups during the practical sessions. Hence the persons who did not enroll in the MOOC, could have paired with others who did, and never needed to.

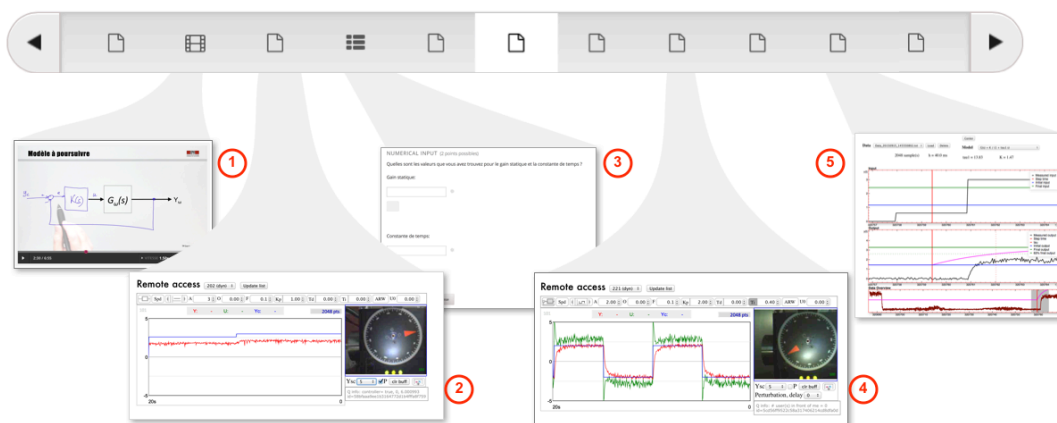


Figure 4.1 – A typical lesson or module structure with consecutive tabs on the upper horizontal strip, each comprising a different type of learning resources: a video, quiz, CPL web app and other.

4.2. The Control Systems Lab MOOC

<i>Course</i>	<i>Num. of students enrolled in course</i>	<i>Num. of students enrolled in MOOC</i>
Automatic Control	110	85 (77.27%)
Control Systems	99	78 (78.78%)
Total	209	163 (77.99%)

Table 4.1 – Enrollment in the Automatic Control and Control Systems courses vs. the enrollment in the MOOC

4.2.2 Remote access

The electrical servo drive is integrated in dedicated tabs of a module's sequence. The web app is shown in Figure 4.2, through which students push their parameters to the lab using the control strip (element #1 in the mentioned figure), they can see a graph of the collected measurements (element #2), and a live video of the status of the motor (element #3). The students can save their experimental results, which are the applied parameters and a data capture of the corresponding graph. Later, they can load them in the available tool for system modeling and interactive simulation (element # 5 in Figure 4.1).

The complete laboratory infrastructure was designed with scalability in mind [62]. Three approaches were devised to handle potentially massive concurrent access to the labs. The first being multiplying the number of CPL setups, as a result the infrastructure services 25 identical installations of the servo drives. The second approach is to implement two roles which users can take: an *observer* who can only watch what the *controller* is doing when a given installation is being used. The third approach is to pre-allocate a fixed time for experimentation when *observers* are queuing for access. Users were allocated by a master load balancer servo CPL, which orchestrates access in a round-robin fashion. When a user connects, s/he are directed to an available installation. If all installations are in-use the users attempting to connect are queued. Students can also choose which setup to use through a numbered list of the available devices (item #4 in Figure 4.2).



Figure 4.2 – The servo drive lab web app user interface composed of a command strip (1), an oscilloscope graph (2), a real-time video feed transmitting the action of the motor (3), and a drop down menu to choose a specific CPL web app from the farm of 25 setups.

4.3 Data Collection Framework

In this section, we describe the Control Systems Lab MOOC data collection framework, including the time frame, the data types collected and an analysis of any data privacy concerns pertaining to the subjects of this study.

4.3.1 Duration of Collection and Data

The course ran from the 20th of September to the 23rd of December 2016, for the exception of some weeks, the students were required to go to the physical lab room twice a week. The data collection happened between October 25, 2016 and December 22, 2016; coinciding with 12 days where lab sessions were scheduled at the premises.

For every connection to a servo drive CPL setup, a one-line trace is recorded containing information regarding the connection, the user and the actions associated with that trace. A complete trace has 25 parameters, 22 of which are of interest to our study and they are listed in Table 4.2. The 3 remaining parameters are related to the system administration for safekeeping the setups. Building and saving the traces is triggered by the incoming requests to lab setups. As it is the current design of the user web app, it pulls for update information from the lab server every 250 ms, which results in many duplicate traces. The total number of collected traces is 954 611, after removing incomplete traces, i.e. traces which were corrupted, the number of these traces is 954 551. We remove duplicate traces, which reduces the total to 63 885 traces, including those generated by system administration staff for testing. When running the analyses, we dynamically remove the traces coming from testing users identified through the **uID** parameter, described in Table 4.2.

4.3. Data Collection Framework

<i>Parameter</i>	<i>Description</i>
user IP and port	IP and port from which the user is connecting to the lab
timestamp	time at which the user exerted an associated action
AM/PM	time of the day the hour timestamp belongs to
date	date of the associated trace
role	controller or observer
smart device IP and port	IP and port of the Smart Device serving the connection
queue	queue size in which the current user is in for access
12 experimentation parameters	parameters which users can command the process with
uID	edX anonymous user id
experiment id	identifies which tab of experiments the lab is being used from
allocated time	preset duration for the lab in seconds

Table 4.2 – Parameters used in the traces and their descriptions

4.3.2 Privacy Impact Assessment

Online user's privacy is a trending topic, especially in educational online environments. In this section, we devise a privacy impact assessment report (PIA) to address any privacy related issues which might arise. The PIA is a measure of our ability to keep private information safe by ensuring conformance with applicable legal and regulatory data privacy policy requirements, determining the risks and effects of our study on the subjects (in our case the students), and evaluating protections and alternative processes to mitigate potential privacy risks [10, 82]:

A) The need for a PIA: it is recommended to draft a PIA when a data controller (i.e. the entity deriving value from the data) is in possession of personal identifiable data such as names and ages of the subjects, and when the initial purpose of the data collection has changed. The data used for the studies in this thesis originates from two data sources: the Open edX database which has all personal identifiable information of the subjects, and the tracking implemented on the servo drive CPL side. The latter source only has access to the **uID** of a student, and on its own cannot be used to link any of the collected traces to a single individual. As to the initial purpose of the data collection, it has indeed changed since the start of the study. Originally, we were solely interested in the **access** to the servo drives CPLs, and the harvested data originated only from the lab server side. As more value could be derived from the insights, we acquired the Open edX data for user matching and performance studies. Even though the initial purpose of the data collection and analysis changed, and aggregation of new data was done, we are still withing the law of data privacy and protection as per the Article 13 of the FADP (Federal Act on Data

Chapter 4. Computational Analysis of Students' Access and Use of CPLs

Protection ¹⁾ [9, 12] and to the Terms of Service of Open edX.

B) Institutional and governmental guidelines: two main guidelines should be followed: first the FADP which is made public and is very detailed on the Swiss government website, and second the Open edX Terms and Conditions (referred to as the General Usage Conditions). These two texts help in identifying and mitigating any data privacy issues or conflicts which would arise in our case.

C) Data flow: the Open edX data is collected and kept by the CEDE (Center of Digital Education - MOOC Factory) group at EPFL. At our request, the data was shared through a secure institutional link after an NDA was signed. After reception of data, the link was rendered inactive and we stored the data on a location inaccessible through an internet connection and unattainable but explicitly by the data controller. This is the portion of the data which contains the personal identifiable information. The data collected from the individual CPLs is sitting on a secured institution server, only accessible to the data controller. This data is anonymized and is useless to link any action to a specific individual and does not contain any personal or sensitive information.

D) Privacy risks: sensitive data or information could cause privacy risks. According to Article 23 of the FADP, sensitive data is defined as data related to the following: (i) religious, ideological, political or trade union-related views or activities; (ii) health, the intimate sphere or the racial origin; (iii) social security measures; (iv) administrative or criminal proceedings and sanctions. Hence, our data is not considered sensitive data, and all rules which apply to it do not for our data.

E) Subjects consent: students explicitly consent to the data collection and further processing when they sign up on Open edX, according to Article 1.2 of the General Conditions which they accept [13] before starting the MOOC. Even though it is a requirement to the use of the platform, and it could be 'ethically' argued that the subjects were forced to accept these conditions, Swiss federal law on data privacy and protection [9, 12] resolves this conflict according to Article 13, section 1, of the FADP which states that is legal for the data controller to use the data if it: "*Processes personal data for purposes not relating to a specific person, in particular for the purposes of research, planning and statistics and publishes the results in such a manner that the data subjects may not be identified*".

¹<https://www.admin.ch/opc/en/classified-compilation/19920153/index.html>

4.4. Descriptive Statistics of Access to the Servo Drive CPL

<i>Exp. Parameter</i>	<i>Encoding</i>		<i>Exp. Parameter</i>	<i>Encoding</i>
S_Mode	0		P_Kp	6
S_Input	1		P_Ti	7
S_Frequency	2		P_Td	8
S_Shape	3		P_ARW_ON	9
S_Amplitude	4		P_U0	10
S_Offset	5		W_Delay	1

Table 4.3 – Parameters encoding table. The symbols are used in the sequences for processing.

4.3.3 The Sequences Dataset

The sequence dataset SD is the list of sequences, where a sequence in this context is an ordered list of symbols each encoding the parameter the student altered when experimenting. Let $S = \{p_1, p_2, \dots, p_m\}$ be a set of parameters or items in the sequence S .

A sequence (also an experimental attempt) S is a list of parameter changes a student does for an uninterrupted duration of 90 seconds. Each trace is compared to the one preceding it, if the elapsed time is less than 90 seconds, then the parameter change is added to the sequence. If it's more than 90 seconds, a new sequence is initialized. The threshold of 90 seconds was chosen based on the need, for some experiments, to wait for around 90 seconds to have the whole graph filled with points (element #2 in Figure 4.2).

Using the CPL web app, students can alter 12 parameters, listed in Table 4.3. The set $\{5, 4, 6, 5\}$ is an itemset containing 4 actions the students did during one experimentation session. In the set of items $\{5, 4, 6\}$, the parameter changes 5 occurred first, followed by parameter change 4 and then 6.

4.4 Descriptive Statistics of Access to the Servo Drive CPL

The resource allocation scheme presented in Section 4.2.2 was devised to accommodate massive concurrent access and support off-campus access to the lab installations. In this section, we analyze the **access** to the CPLs: from where students are connecting (on or off-campus), how much time they are spending doing experimentation, how much they queued for the access and how concurrent access manifested on the lab farm.

4.4.1 Location of connection

We check for unique user IPs which connected to the servo drive CPLs, from which we can tell where the students were located: on or off-campus. Figure 4.3 plots the distribution of the connection locations. We can see that 10.71% (35 distinct computers) of the connections were made from the lab room where students can see the equipment. Around 8.8% came from the EPFL campus network, around 61% from the either the EPFL wifi network or remote VPN access, and 19.51% from outside any EPFL provided connection.

The fact that the majority of the connections are from either the EPFL wifi or VPN network suggests that students are not limiting themselves to the lab hours to do the experiments, or that they are bringing in their own devices to the lab hours. In Sections 4.4.2 and 4.4.3), we explore the dates and times where students were connected to the labs and how they queued to reveal the specifics of these patterns.

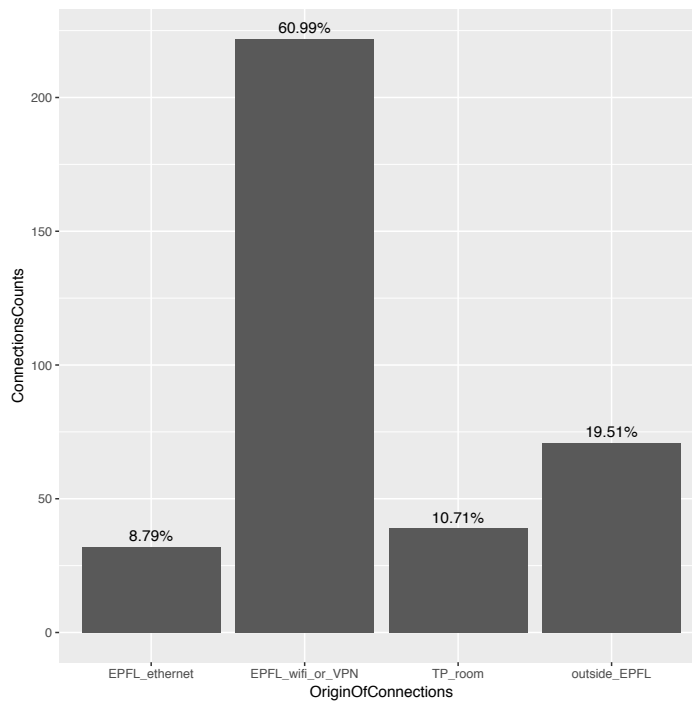


Figure 4.3 – Distribution of the origins of connections to the CPLs

4.4.2 Queue Sizes

When students try to connect to a setup which is already occupied or when all setups are occupied, they are pushed to a first-in-first-out queue. The maximum queue size encountered by students is 3 (0.1% of the total connections), a queue of 2 was observed for 2.89% of the connections, and 97% of connections were for non-queuing users. When removing all students who spent less than 10 seconds connected, we don't observe any more queuing, and the 25 setups are successfully servicing all requests to the CPLs setups. More specifically, no students waited more than 1 second in a queue, and queuing is occurring only during the peak time for connections, which mostly corresponds to the time of the scheduled lab sessions. The fast drop out of the queue can be explained by the fact that, maybe students who are gaining control of the setup are leading the group work, and others are disconnecting or switching to other tasks such as video watching. The CPL farm is hence not saturated.

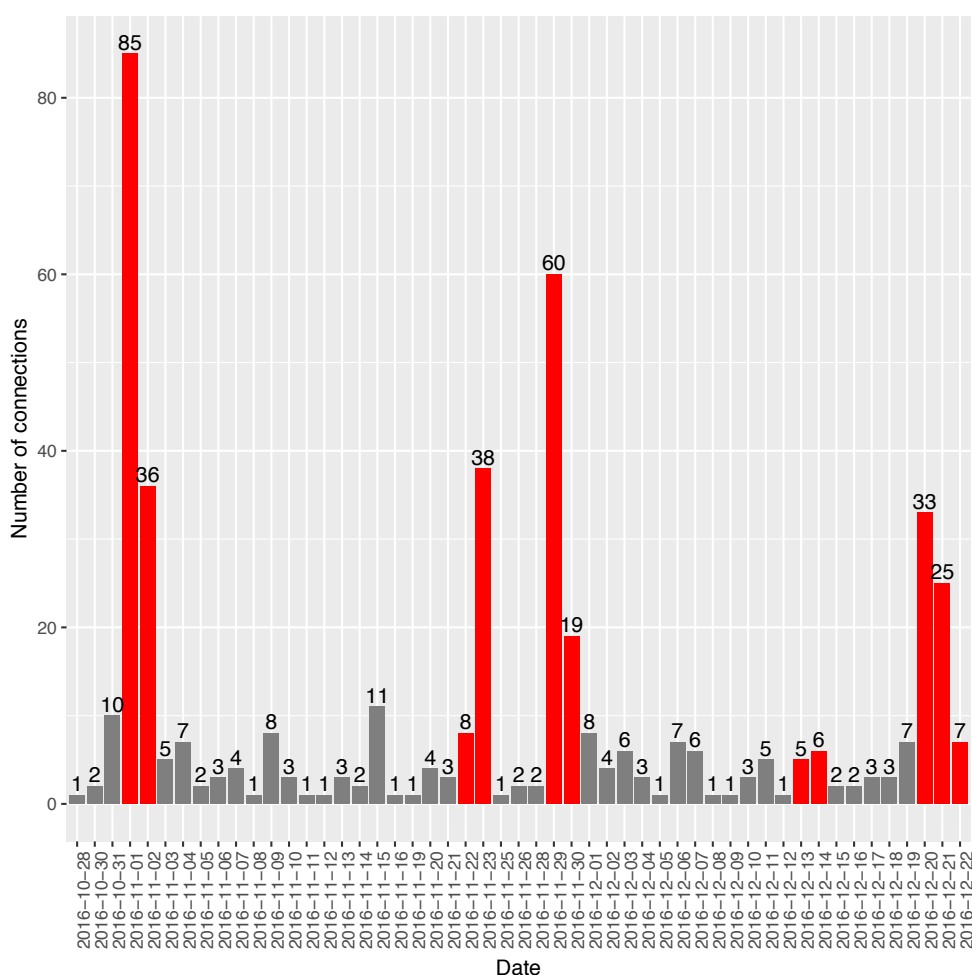


Figure 4.4 – Number of connections per day, on the days where the CPLs were accessed. The red bars correspond to the days of the scheduled lab sessions at the university.

4.4.3 Concurrent Access

In Figure 4.4, we show all the days during which the students accessed the MOOC. The CPLs were accessed on 51 days between October 25 2016 and December 22 2016. Scheduled lab sessions took place on 11 of the 51 days. The dates of the scheduled lab sessions are: November 1, 2, 22, 23, 29, and 30, December 13, 14, 20, 21, and 22. This shows that students are not only interested in the material offered by the course, but given the opportunity they will take it in order to experiment. The maximum number of connections per day is 85, the minimum is 1, the mean is 17.75 and the standard deviation is 36.84.

4.4.4 Duration of experimentation

Figure 4.5 shows the total time spent by students experimenting when they connect to the MOOC on a day. The maximum cumulative time spent on experiments by students in a day is 8 hours 7 minutes 3 seconds, and the minimum is 56 seconds. Notice that the CPLs are used mostly on days when lab sessions are scheduled (red bars). Similar trends can be seen for days where there were no scheduled lab sessions (grey bars).

Based on the raw data collected from all the lessons, we found that the minimum connection time is 1 second, the maximum is around 1 hour, the mean is 6 min and 22 seconds, with a standard deviation of 11 min and 29 seconds. Table 4.4 shows more granular statistics regarding the mean, standard deviation of experimentation time spent by the students in each module, in addition to the pre-allocated time to experiments, the number of experimentation tabs, and the allocated time per tab for each module. We notice that the mean and standard deviation of the experimentation time in all the course modules are close, regardless of the shortest allocated time (30 seconds) or the largest (4 minutes) per module, as well as of the number of experimentation tabs per module.

<i>Module</i>	<i>Mean</i>	<i>StdDev</i>	<i>AllocatedExpDuration</i>	<i>NumExpTabs</i>	<i>AllocatedDurationTab</i>
Intro	08:00	11:51	00:30	1	30
Module 1	09:50	13:09	01:00	2	30
Module 2	06:13	10:31	04:30	3	90
Module 3	06:15	11:43	08:00	4	120
Module 4	07:37	12:35	08:00	4	120
Module 5	06:52	11:25	04:00	2	120
Module 6	04:33	08:58	06:00	3	120
Module 7	06:27	11:04	04:00	2	120

Table 4.4 – Mean, standard deviation, and allocated time in minutes:seconds format, number of experimentation tabs, and allocated duration per experimentation tabs in seconds, for each module.

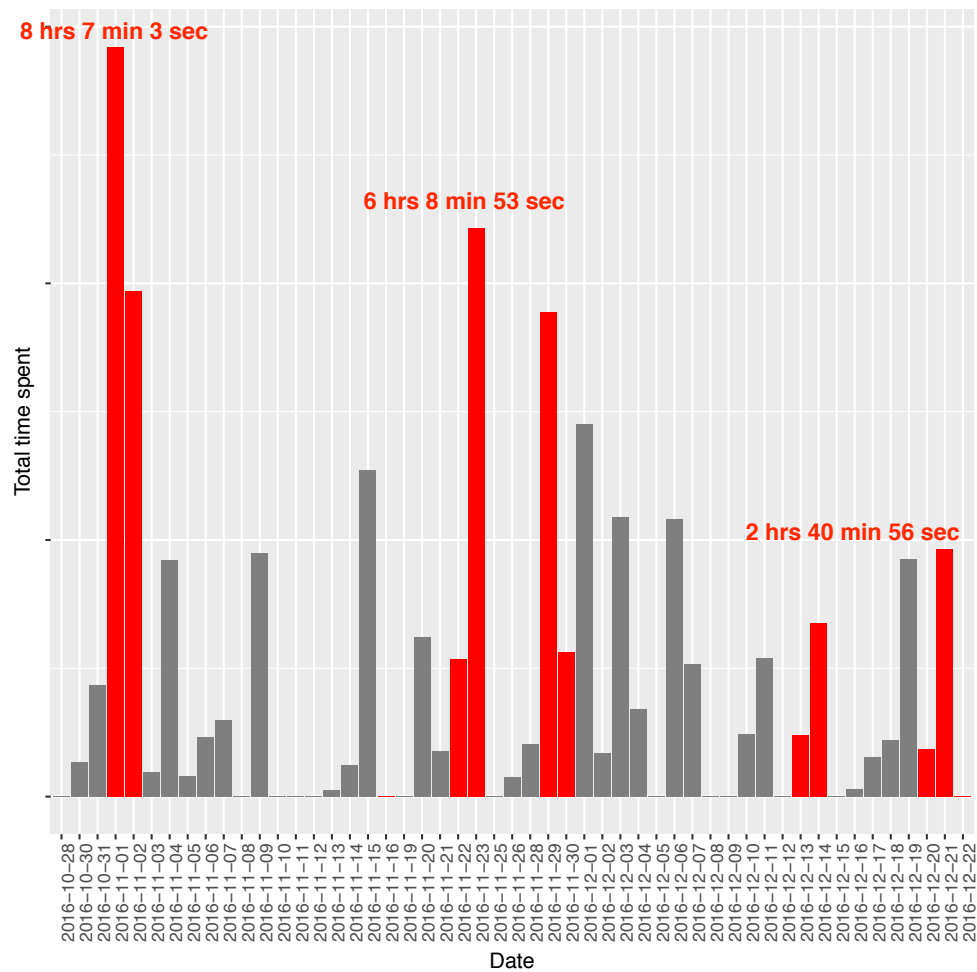


Figure 4.5 – Time spent on experimentation on the days where the CPLs were accessed through the whole period of the data collection. The red bars correspond to the days of the scheduled lab sessions at the university.

4.5 Descriptive Statistics of Experimental Behavior

In this section, we check the descriptive statistics of the students' experimental behavior through the use of the CPLs, then we list the findings and insights.

4.5.1 Analysis of participation

Referring to Table 4.1, we see that 131 of the 163 enrolled students used the labs at least once, contributing to a participation rate of 80.4%. In Table 4.5, we show the average number of times students attempt an experiment, and the corresponding standard deviations. We notice that on average, students are attempting less than once an experiment.

Chapter 4. Computational Analysis of Students' Access and Use of CPLs

<i>ExpID</i>	<i>Mean Numb. Attempts</i>	<i>Std. Dev.</i>		<i>ExpID</i>	<i>Mean Numb. Attempts</i>	<i>Std. Dev.</i>
e1	0.086	0.45		e11	1.87	2.92
e2	0.08	0.33		e12	0.67	1.76
e3	0.22	0.92		e13	0.12	0.49
e4	0.64	1.47		e14	0.68	1.61
e5	0.19	0.6		e15	1.02	1.71
e6	0.41	0.93		e16	0.97	1.76
e7	0.79	1.43		e17	1.22	2.26
e8	0.76	1.40		e18	0.53	1.19
e9	0.57	1.19		e19	0.48	1.09
e10	0.87	1.65		e20	0.45	1.18
				e21	0.21	0.56

Table 4.5 – Participation table showing the average number of times students attempted an experiment.

<i>ExpID</i>	<i>Suggested seq. length</i>	<i>Allocated time</i>		<i>ExpID</i>	<i>Suggested seq. length</i>	<i>Allocated time</i>
e1	4	30 sec		e11	3+	120 sec
e2	4	30 sec		e12	7+	120 sec
e3	2	30 sec		e13	8+	120 sec
e4	4	90 sec		e14	9+	120 sec
e5	3	90 sec		e15	8+	120 sec
e6	4	90 sec		e16	3+	120 sec
e7	1+	120 sec		e17	3+	120 sec
e8	7	120 sec		e18	5+	120 sec
e9	5+	120 sec		e19	9+	120 sec
e10	8+	120 sec		e20	6	120 sec
				e21	4	120 sec

Table 4.6 – The number of steps in the instructor-suggested sequences and the allocated time for each experiment.

4.5.2 Analysis of activity

We describe how active students are during their experimentation activity by analyzing two characteristics of the activity: a first characteristic is the number of actions per sequence, and the second is the duration or the total time spent in those individual sequences sequences.

Sequences length

For each of the 21 experiments of the MOOC, the instructor suggests a sequence of steps for students to follow in their experimentation. Table 4.6 shows the number of steps suggested by the instructor for each experiment. In the table, a *numberOfSteps+* entry means that students need to vary some parameters in an iterative process, and that the number of steps is the minimum one to complete an experiment.

In Figure 4.6 and Figure 4.7, we show the distributions of the number of actions executed by distinct students through boxplot visualizations, for the 21 experiments of the course, denoted with en for experiment n . Considering the suggested number of steps for each experiment as the minimum that a student has to do in order to complete the attempt, we notice that it is not the minimum depicted in all boxplots. For all experiments, the minimum is 0 which is attributed to students who experimented at least once during the course, but did not attempt a given experiment. Yet, for all experiments, the minimum number of suggested actions lays in the second and third quartiles of the boxplot, indicating that it is relatively close to the value which separates the range of number of actions students are doing in a sequence (i.e. the median). Examining the spreadability of the distribution, we see that the interquartile range is between 10 and 38 actions, with the presence of some outliers. More precisely, for the group of students whose number of actions per sequence were closest to the median, half were within 10 to 38 actions of each other when they experimented. Notice that the distributions are sometimes skewed to the left or to the right, and others are symmetric. In this sense, for some cases the distribution leans towards higher number of actions as difference between sequence lengths, in other cases it's less.

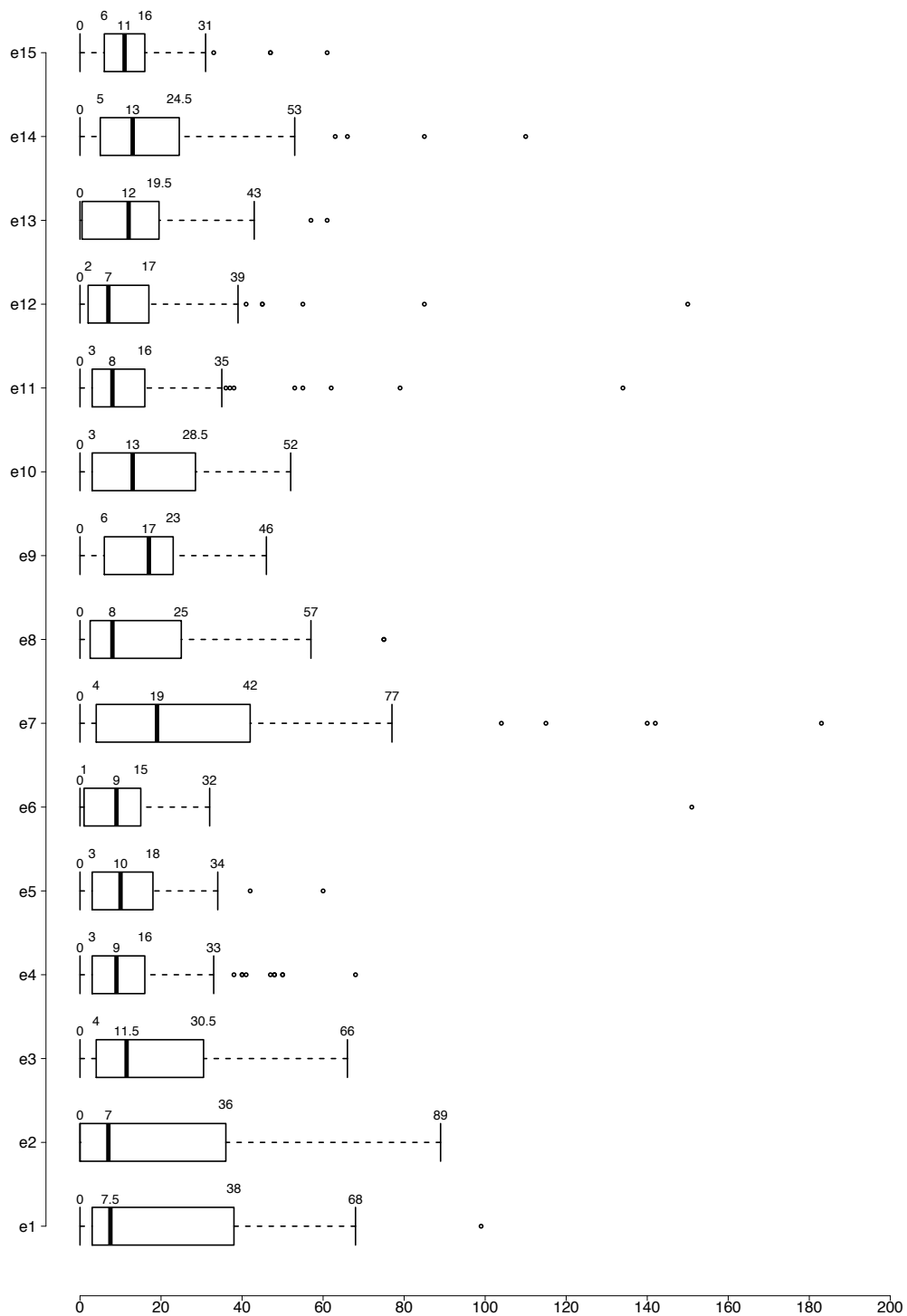


Figure 4.6 – Boxplots showing the distributions of the number of actions exerted per sequence (experimental attempt), for experiments 1 through 15, denoted by e1 through e15 respectively.

4.5. Descriptive Statistics of Experimental Behavior

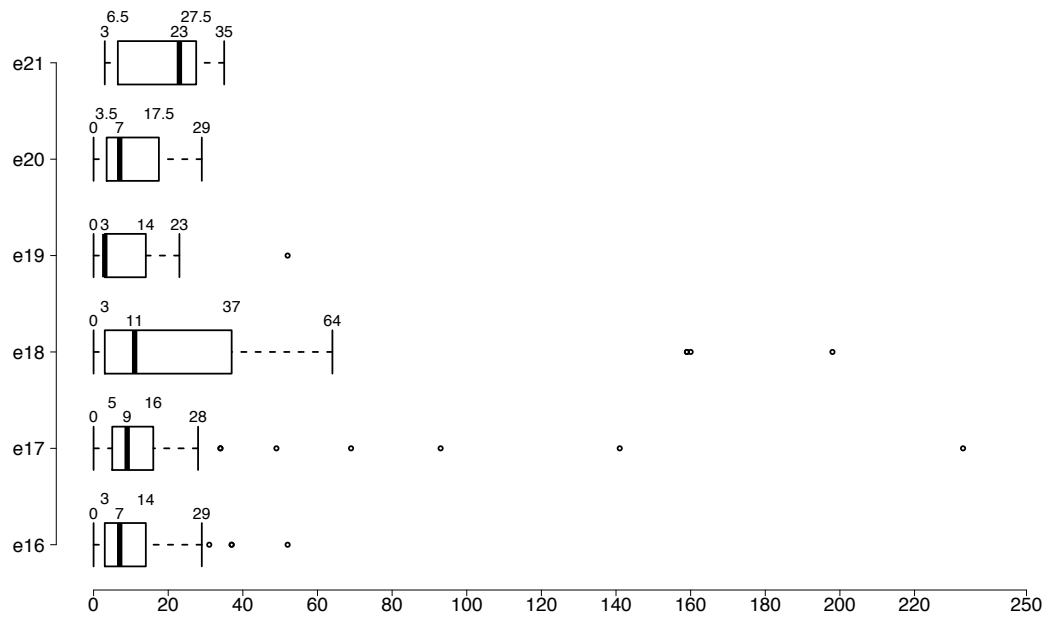


Figure 4.7 – Boxplots showing the distributions of the number of actions exerted per sequence (experimental attempt), for experiments 16 through 21, denoted by e16 through e21 respectively.

Sequences duration

The second activity-related characteristic we inspect is the duration spent in the sequences, or in other words, how much time students spent executing the parameter changes while experimenting. Recall that students can spend as much time as they desire on an experiment, as long as there are no queuing users for the setup in-use. When queuing starts, the current user is permitted a limited time to experiment, after which they are paused and queued. Table 4.6 shows the allocated time for each of the 21 experiments in the case of queuing. These durations were specified by the course instructor.

In Figure 4.8 and Figure 4.9, we show the distributions of the time spent by distinct students through boxplot visualizations, for the 21 experiments of the course. As in the previous figures, experiment numbers are denoted as follows: *en* for experiment *n*. At a first look, we notice that the distributions don't show trends similar to those of the actions boxplots in the previous section (Figures 4.6 and 4.7). In fact, very few boxplots distinctively show a core (the box which encompasses the second and third quartiles). This does not necessarily imply an absence of the boxes, but that the distribution is very heavily-tailed that we cannot see the box without further processing. For all experiments, the median duration is 0 seconds, and all are skewed to the right.

Recall from Table 4.5, that we found low participation rates for students in experiments (all experiment attempt means are less than 1). Not participating in an experiment by a given student contributes to a 0 seconds entry, and given the low participation rate, 0 seconds entries are much more numerous than non-zero seconds entries for the durations spent in an experiment. Hence, to further process the distributions in search for trends, we remove all 0 seconds entries for each experiment. The corresponding distributions are plotted in Figure 4.10 and Figure 4.11. The resulting boxplots show identifiable boxes for distributions, with much less outliers than in the case of Figures 4.8 and 4.9. Considering the allocated time to complete an experiment in case of queuing as the minimum required time to complete an experimental attempt, we see that the medians for the time spent in each experiment are not close to these values, except for the case of experiment 1 (e1, 47.5 seconds for 30 seconds allocated) and experiment 13 (e13, 2 min 28 seconds for 2 minutes allocated). The minimum time spent ranged between 1 and 21 seconds, medians ranged between 40 seconds and 12 minutes 10.5 seconds. Looking at the spreadability of the distribution, we see that the interquartile range is between 1 min 43 seconds and 21 minutes 50 seconds, with the presence of some outliers. More precisely, for the group of students whose time spent experimenting per sequence were closest to the median, half were within 1 min 43 seconds to 21 min 50 seconds of each other when they experimented, for experiments. The distributions are skewed to the right, meaning that the tail of the distribution tends to the right of the axis. The skewness of the distributions to the right indicates that they lean towards higher duration difference between sequence lengths.

4.5. Descriptive Statistics of Experimental Behavior

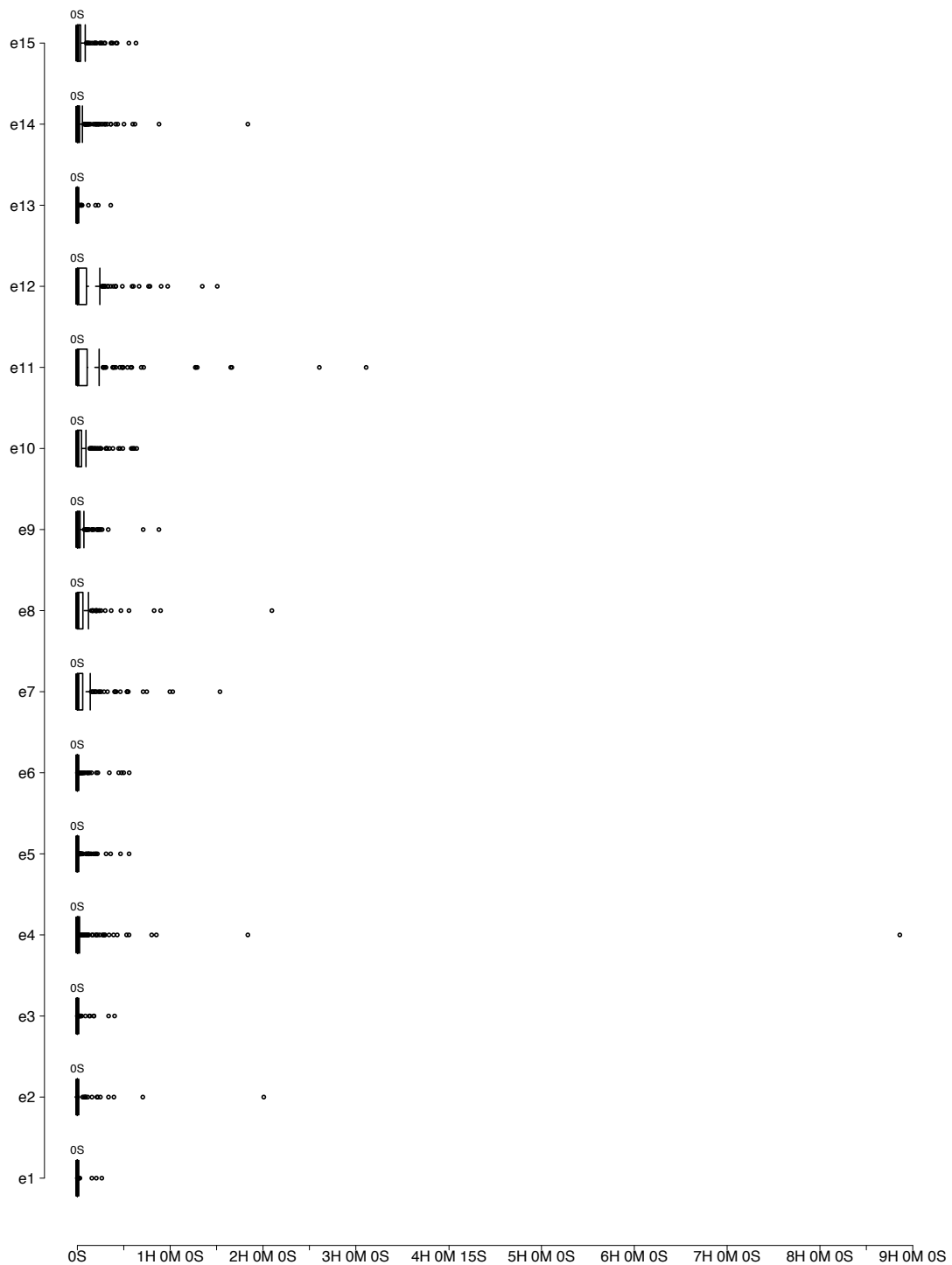


Figure 4.8 – Boxplots showing the distributions of the time spent in experiments 1 through 15, denoted by e1 through e15 respectively. Only the median of the distribution is shown.

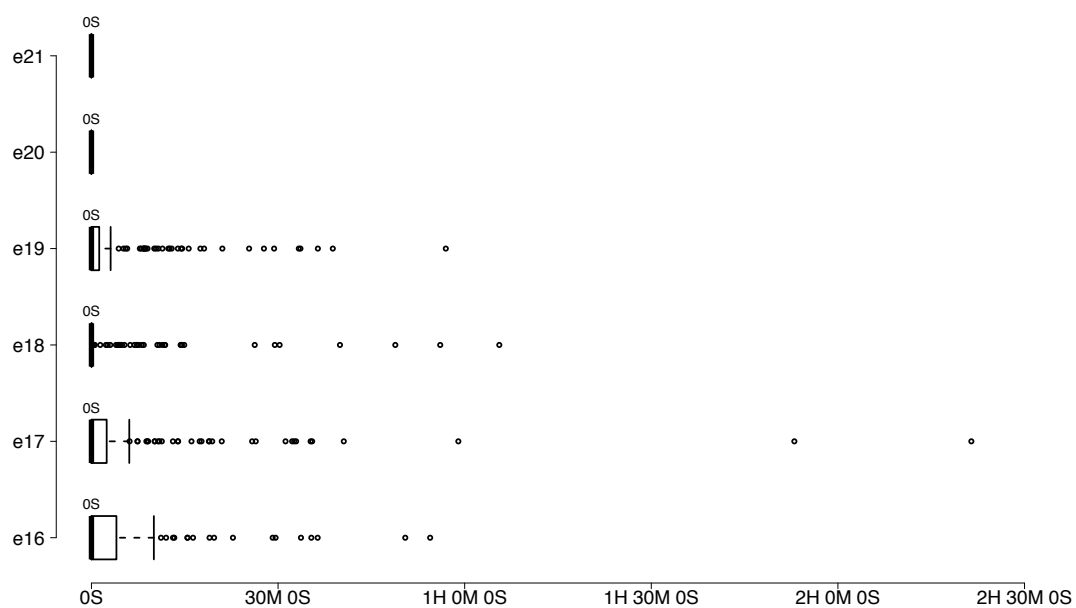


Figure 4.9 – Boxplots showing the distributions of the time spent in experiments 16 through 21, denoted by e16 through e21 respectively. Only the median of the distribution is shown.

4.5. Descriptive Statistics of Experimental Behavior

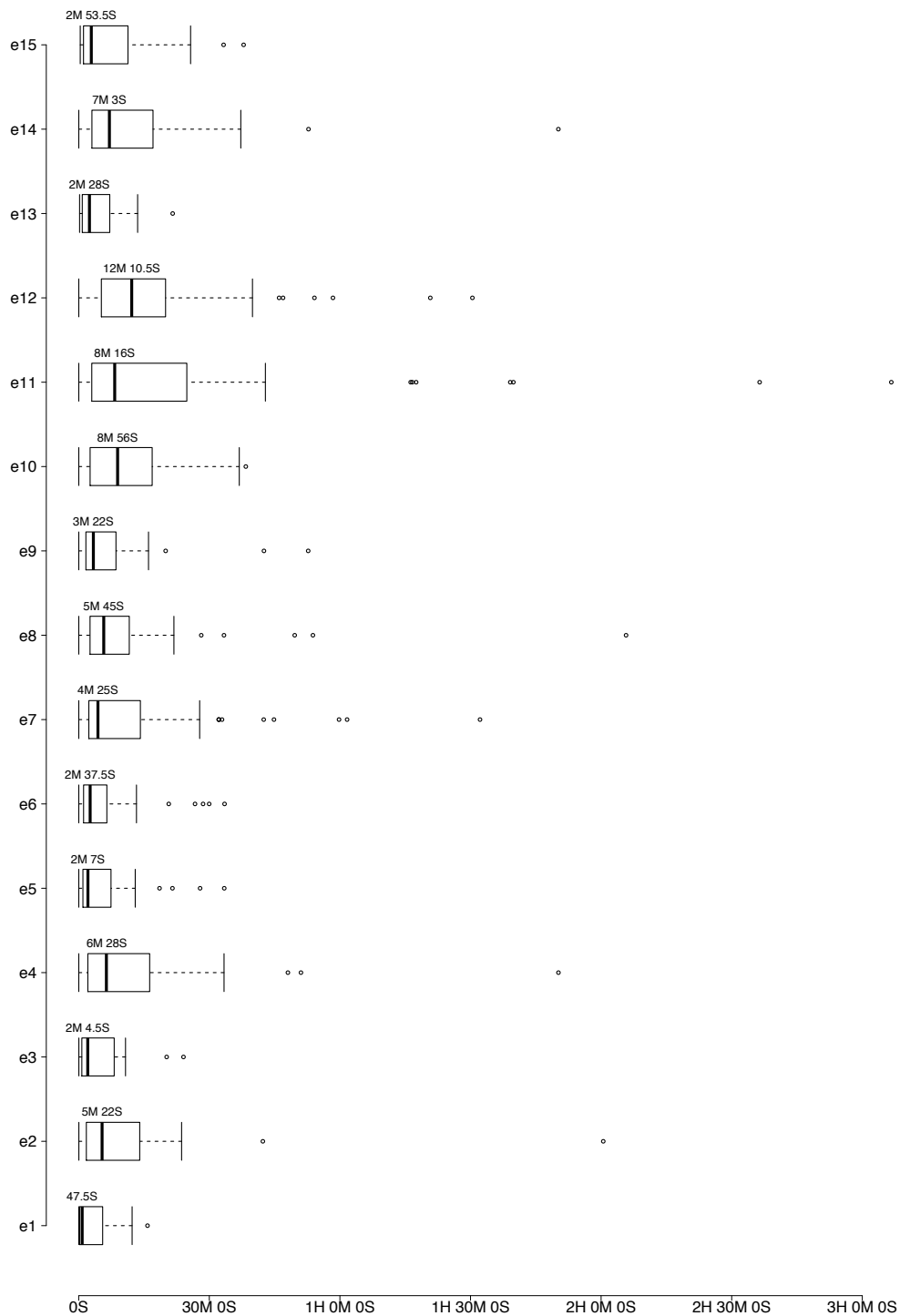


Figure 4.10 – Boxplots showing the distributions of the time spent in experiments 1 through 15, denoted by e1 through e15 respectively, after filtering out 0 seconds entries.

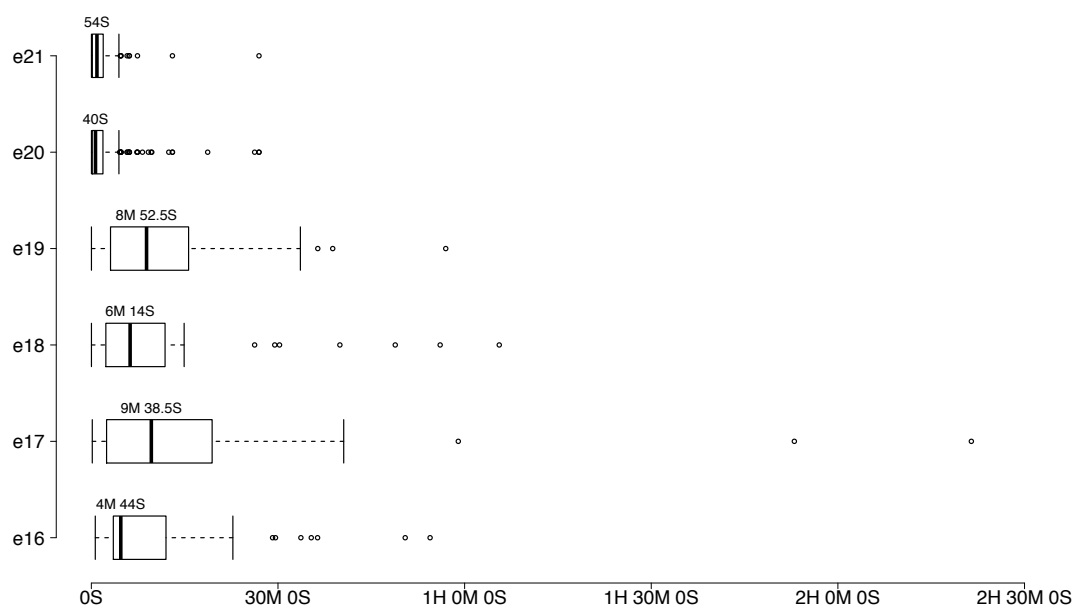


Figure 4.11 – Boxplots showing the distributions of the time spent in experiments 16 through 21, denoted by e16 through e21 respectively, after filtering out 0 seconds entries.

4.5.3 Discussion

To analyze the experimental behavior of students, we inspected two characteristics of the activity: the number of actions or parameter changes in an experimental sequence, and the time spent executing these actions. The distributions of the number of actions in sequences showed no specific patterns, some of the distributions are skewed to the left and others to the right. The number of parameter changes suggested by the instructor lies in the interquartile distance, suggesting that it was enough for students closest to the median of the distribution to complete an experimental attempt.

The distributions of the corresponding sequences durations don't show patterns similar to the ones depicted for the number of actions (Figures 4.6, 4.7, 4.8 and 4.9). The durations distributions are all heavily-tailed, with some showing the body of the boxplots. By comparing the distributions for the number of actions and the corresponding ones for the durations, we can see that they don't show similar trends: not all durations distributions show a body for the box of boxplots, skewness to the right for the number of actions doesn't correspond to skewness to the right in the durations distribution, similarly for left skewness. After removing the zero-valued durations, we re-plot in search for patterns. In Figures 4.10 and 4.11 we see that the distributions show patterns that resemble those of the actions distributions. Yet, all the distributions in this case are skewed right, and there are no proportional or linear relations which can be concluded. In other words, doing more parameter changes in a sequences does not imply a longer time spent in the corresponding sequence.

The presented statistics could reveal indicators of certain learning behavior traits such as perseverance, dedication, consistency, organization, curiosity, participation and others. For instance, the number of times a student attempts an experiment is reflected in the number of sequences we gather for that student for a given experiment, which suggests how perseverant the considered student is. A more granular example is by inspecting the number of actions per sequence, and how much time a student spends experimenting, as possible indicators of how active and careful they are, and what are their effects on their academic performance. In the following section, we study the effect of the experimental behavior (time spent experimenting, the number of actions in sequences, abiding by the suggested experimental protocol) exhibited by students on their grades.

4.6 Studying the Effects of Students' Experimental Behavior on Academic Performance

In this section, we look at the data from a different perspective. We group the data according to the grades students scored in the MOOC, those who scored above a certain grade (67% of the maximum grade, also the passing grade) are labeled as high-performing, and the rest are the low-performing group of subjects. We start by detailing the methodology adopted in this section, then we present the results and last we discuss our findings.

4.6.1 Method

To check if different characteristics or factor of students' experimental behavior have an effect on their academic performance, we resort to hypothesis testing [78]. Hypothesis testing is a method of statistical inference, where data groups are compared. The study relies on formulating a hypothesis in the form of a *null hypothesis* denoted by H_0 . A common form of the *null hypothesis* is: “*there are no differences on the statistic between groups on a considered factor*”. The rejection of H_0 leads to the acceptance of an *alternative hypothesis*, denoted by H_1 . The alternative hypothesis H_1 represents the question to be answered or the theory to be tested. The *null hypothesis* nullifies or rejects H_1 , and is often the logical complement to H_1 (H_0 is $\neg H_1$). When H_0 is rejected, we conclude that the groups defined by *factor* are different on the tested *statistic*.

The acceptance or refusal of a hypothesis relies on the p-value (probability) derived for the method of hypothesis testing. The p-value represents the probability of observing the sample when the hypothesis is true. The decision to accept or reject the hypothesis is based on a significant level to be set on the value of the probability, denoted by α . The significant level is a threshold to either accept or reject the *null hypothesis*. A 0.05 level of significance implies that if we find a p-value less than 0.05, we reject the null hypothesis with a 95% confidence. Or in other words, the probability of observing the considered sample when the null hypothesis is true is less than 0.05. In this study, we use $\alpha = 0.05$ as our level of significance.

There are two types of statistical tests to obtain the p-value: parametric and non-parametric. Many factors contribute to the choice between the two. A first factor to consider is the assumption on the samples distributions. The first type assumes that the groups are taken from normal distributions, and they test for groups' means. Non-parametric methods do not assume anything about the distributions from which the groups are drawn, and they test of the groups' median. A second factor is whether or not repeated measures are present in the groups. A third factor to take into consideration is the number of groups. Typically, parametric tests are conducted when 2 groups are compared. Other factors specific to a selected parametric method should also be considered appropriately.

We check for the possibility of using ANOVA (Analysis of Variance) [65] as a parametric test.

4.6. Studying the Effects of Students' Experimental Behavior on Academic Performance

We plot the distributions of the total duration spent by students experimenting throughout the course, the total number of experimental attempts (sequences), and the total number of actions they exerted on the system (Figures 4.12, 4.13, and 4.14), by dividing the data into two groups: high-performing and low-performing students. Even though one of the assumptions for ANOVA is data normality, and our data shows skewness of distributions, it is known that the method performs well with skewed data. The second assumption is that data is independent, while our samples have repeated measures (multiple sequences for the same student). The third is that the data have homogeneous variance across groups. The ANOVA assumption on homogeneity of variance across groups is violated, all Laveane's p-values < 0.01 . Hence, we resort to hypothesis testing using non-parametric methods.

We select the Kruskal-Wallis [46] test to conduct our study. The Kruskal-Wallis test compares the groups by their medians. The corresponding null hypotheses H_0 are *there is no difference on medians of the total duration spent experimenting, the number of actions in sequences, and the number of experimental attempts, between high and low-performing students*. In other words, the low and high-performing students when considered on their total time spent experimenting, number of actions exerted in experimental sequences, and number of experimental attempts, show similar or are drawn from the same distributions.

The p-value obtained from the Kruskal-Wallis test reveals overall statistically non-significant or significant differences between group medians. In the later case, when the p-value < 0.05 , to confirm between which groups the differences occurred, a post-hoc test is conducted by getting the effect size (correlation coefficient) [50]. In this study, there are only two groups, hence having a significant result is enough to confirm the differences. But to quantify the magnitude of the difference, the post-hoc test is used to calculate the effect size. The absolute value of the effect size ranges between 0 and 1, values closest to 0 indicate low effect sizes, values closest to 1 reveal high effect sizes. In our study, we follow Cohen's rule of thumb for the interpretation of the value of an effect size: an effect size lower than 0.2 as small, between 0.2 and 0.5 as medium, and larger than 0.5 as big. Since repeated measures are present in our sample (same students had multiple sequences), the adequate post-hoc procedure is the Wilcoxon Rank Sum Test [81], from which we get the z-value to use in the formula for the effect size:

$$r = \frac{z}{\sqrt{N}};$$

where z is the z-value obtained from the Wilcoxon Rank Sum Test and N is the total number of samples.

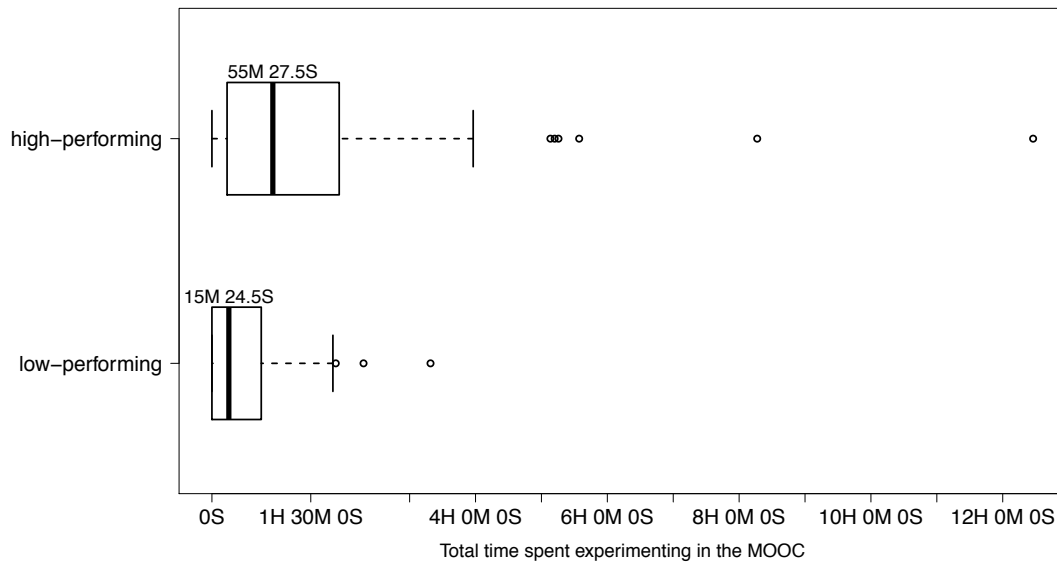


Figure 4.12 – Boxplots showing the distributions of the total time spent by low and high-performing students experimenting, throughout the whole duration of the MOOC.

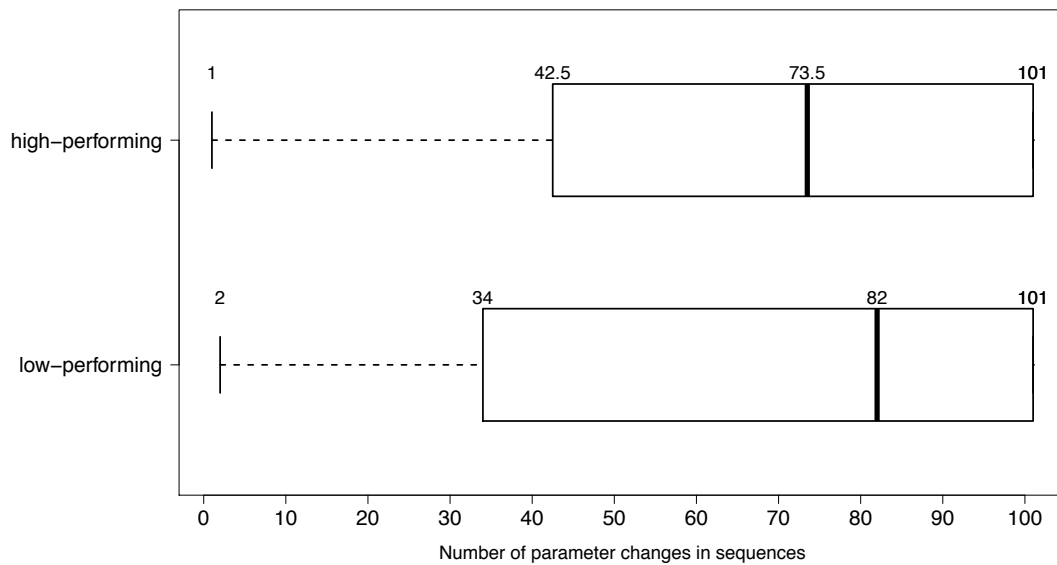


Figure 4.13 – Boxplots showing the distributions of the number of parameter changes for attempts (sequences), for low and high-performing students.

4.6. Studying the Effects of Students' Experimental Behavior on Academic Performance

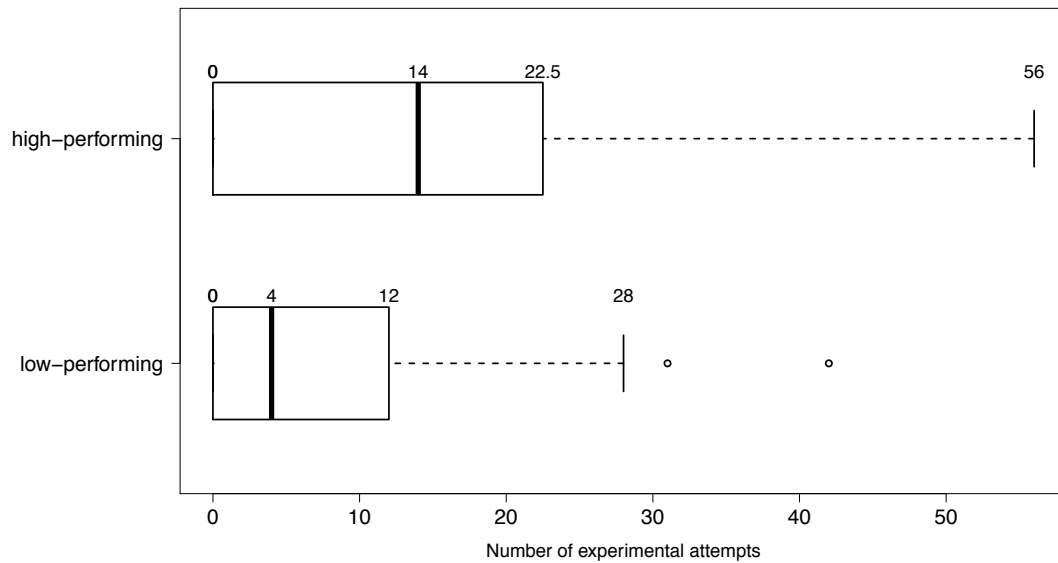


Figure 4.14 – Boxplots showing the distributions of the total number of experimental attempts for low and high-performing students, throughout the whole duration of the MOOC.

4.6.2 Results

The Kruskal-Wallis test provided sufficient evidence to reject the hypothesis that the **total duration spent experimenting throughout the course** and the **total number of sequences (experimentation attempts)** ($p < \alpha = 0.05$) are drawn from a similar distribution for low and high-performing students. A post-hoc test with Wilcoxon rank sum, showed an effect size of $r = 0.3$ and $r = 0.22$, for the total duration spent experimenting throughout the course and the total number of sequences respectively.

The number of actions per sequences and to which academic section the students belonged does not show a significant overall difference. All the results are tabulated in Table 4.7.

<i>Test statistic</i>	<i>p-value</i>
total duration spent throughout the course	p-value = 0.0001605*
number of parameter changes per sequence	p-value = 0.595
total number of experiment attempts (sequences)	p-value = 0.000595*
class section	p-value = 0.252

Table 4.7 – The p-values for the Kruskal Wallis test for the total duration spent experimenting throughout the course, the number of parameter changes per sequence, the total number of experiment attempts, and class section.

4.6.3 Discussion

When plotting the distributions of the total time spent experimenting by low and high-performing students throughout the course in two separate distributions, we see in Figure 4.12 that high-performing students tend to spend more time experimenting (median = 55 minutes 27.5 seconds with a long tail to the right of the distribution) than low-performing students (median=15 minutes 24.5 seconds and a shorter right tail). A statistical significance test with the Kruskal-Wallis method revealed a significant difference between the two distributions, indicating that the total time spent during the course experimenting is drawn from different distributions for low and high-performing students. However, the effect size of this difference (detailed in Section 4.6.1), or its magnitude is medium ($0.2 < r = 0.3 < 0.5$).

The distributions for the number of parameter changes in sequences for low and high-performing students show that low-performing students tend to change slightly more parameters in an experiment (median = 82) than high-performing students (median = 73.5). The Kruskal-Wallis test resulted in no significant difference between the distributions of the number of parameter changes in a sequence for low and high-performing students. Examining Figure 4.13, we notice that the medians of the two distributions only differ by 11 actions for a sequence, they both have a maximum of 101 parameter changes per sequence, the interquartile distance is 67 for low-performing students and 58.5 for high-performing students. The minimum for low and high-performing students' distributions are 2 and 1 respectively. Hence, these two distributions are comparable, with the distribution of the number of parameter changes for low-performing students being spread more than the one for high-performing students by 8.5 actions.

Figure 4.14 shows the distributions of the number of experimental attempts throughout the MOOC for low and high-performing students separately. High-performing students tend to attempt to experiment more (median = 14) than low-performing students (median = 4). A hypothesis test revealed a significant difference between the two distributions with a p-value < 0.05 . The associated effect size is $0.2 < r = 0.22 < 0.5$, indicating that the difference between the two distributions is medium. Notice that for both distributions, the minimum is 0, which corresponds to students who didn't attempt any experiment throughout the course. Hence, even though an overall difference exists between the two distributions (evidence from the hypothesis test leading to a p-value $< \alpha = 0.05$), the effect is small, and this can be explained by the presence of students who didn't attempt any experiment throughout the course in both groups.

P-values, priors, posteriors and causality

The p-value is a measure of discrepancy of the fit of a null hypothesis H to data Y . It is defined as $Pr(S(Y_{new})) > P(Y_{observed}|H)$. Where Y_{new} is a hypothetical replication of data $Y_{observed}$ under the null hypothesis H , and S is the test statistic. In this sense, p-values derived from statistical hypothesis testing are commonly viewed as the posterior probabilities of the correctness of stating that the null hypothesis is false, as more data is observed (long-term frequencies, or the frequentist

perception) [8, 33]. Many argue that this interpretation is misleading [33, 37, 54], and prefer to adopt a Bayesian perspective. In this regard, they claim that p-values derived from hypothesis testing are a lower bound on posterior probabilities under non-informative priors ($P(B|A)$ where $P(A)$ is the prior, also referred to as a quantification of uncertainties approach).

Our study was conducted in an observational setting, where there were no assumptions about the data generation process (priors). In other words, we had no prior knowledge of the academic performance of students in general, or in other material-related courses (high or low), and how they related to the observed trends for the total time spent experimenting, the number of experimental attempts and the number of parameter changes per sequence. Hence, in our study causality (of observing such trends for the tested statistics) cannot be implied without a knowledge of the priors. Causality is based on dynamically changing information, where we would be searching for the probability of observing the same distributions (for time spent experimenting, number of experimental attempts and number of actions per sequence) with a different pool of students, given the characteristics of the students (low or high performing students, known as priors):

“The aim [of causality] is to infer aspects of the data generation process. With the help of such aspects, one can deduce not only the likelihood of events under static conditions, but also the dynamics of events under changing conditions. This capability includes predicting the effect of actions (e.g., treatments or policy decisions), identifying causes of reported events, and assessing responsibility and attribution (e.g., whether event x was necessary (or sufficient) for the occurrence of event y)” [54].

4.7 Mining Students' Experimental Patterns

CPLs present to learners an unstructured learning environment where they can act on the system without constraints on the order and number of steps, an environment referred to as ‘exploratory’. Exploratory learning environments have advantages and disadvantages: on one hand they give students the opportunity to familiarize themselves with the lab at ease, without any stress of space and time so they can experiment as it suits them; on the other hand it is very likely that in such settings students take longer to get to the results they need, or they experiment less effectively.

The data collected from the use of the CPLs considered in this chapter can help in finding out what experimental patterns students are leaving behind, and how they are related to their performance in the course. In this section, we first detail our method for sequence mining to be used in statistical testing, then we present corresponding results.

4.7.1 Method

Sequential pattern mining is a data mining task specialized in discovering patterns in sequential data. The interestingness of a subsequence can be measured in terms of its occurrence frequency in the sequence dataset (frequent sequences or subsequences), length (number of items in a subsequence or sequence) and profit (in our case being academic performance).

We mine for frequent experimental sequences or subsequences for each of the experiments (e1 through e21). A sequence or subsequence S is said to be a frequent sequence or subsequence, if and only if $support(S) > minsup$, where $support$ is for a threshold $minsup$ set by the data controller. For our study, we set the $minsup = 0.3$, searching for sequences and subsequences which accounted for at least 30% of the sequences and subsequences for a given experiment.

Using the chi-square statistical test (a non-parametric test), we identify discriminant subsequences, or in other words the sequences and subsequences which significantly discriminate a group, with the Bonferroni correction method for repeated measures (same student contributing to more than one sequence for an experiment). As in Section 4.6, the groups are the low and high-performing students, and the variable we are testing against are the frequent sequences and subsequences found in a given experiment sequences. The null hypothesis H_0 is: there is no difference between the groups (low and high-performing students), discriminated on the frequent subsequence/sequence. The subsequences are then ordered by decreasing discriminant power, using Pearson's residuals (equivalent to the effect size discussed in Section 4.6.1).

4.7.2 Results

Figures 4.15 through 4.35 show the results of the statistical test discussed in the previous section. The color of each bar represents the associated Pearson residual of the Chi-square test. For residuals calculated at a p-value < 0.01 (significant level is $\alpha = 0.01$ as detailed in Section 4.6.1), the bars are in dark red for a significant negative effect, and dark blue for a significant positive effect. For a p-value < 0.05 , the bars are in orange for a significant negative effect, and in light blue for a significant positive effect. White columns indicate that the statistical test found no difference between the considered groups for the considered subsequences/sequences (accept H_0). A significant negative effect indicates that the subsequence is significantly less frequent (dark red and orange) than expected when the null hypothesis is rejected, while positive effects (light and dark blue), the subsequence is significantly more frequent.

Only the frequent sequences of experiment 18 (Figure 4.31) show significant results. For high-performing students, the use of parameter 6 (of the encoding Table 4.3) is significantly less frequent than expected, while it is significantly more frequent for low-performing students. As for the subsequences $(4>7)$ and $(4>7)-(7-8)^2$, it is significantly less frequent for low-performing

²The notation $(4>7)$ represents the subsequence where parameter 4 followed by parameter 7. A $(4-7)$ represents a subsequence where items 4 and 7 happened, without importance of order of occurrence. This detail is of no interest to our study.

4.7. Mining Students' Experimental Patterns

students. And we can see that low-performing students have exhibited these sequences much less frequently (around 27% frequency) vs. 55% frequency for high performing students.

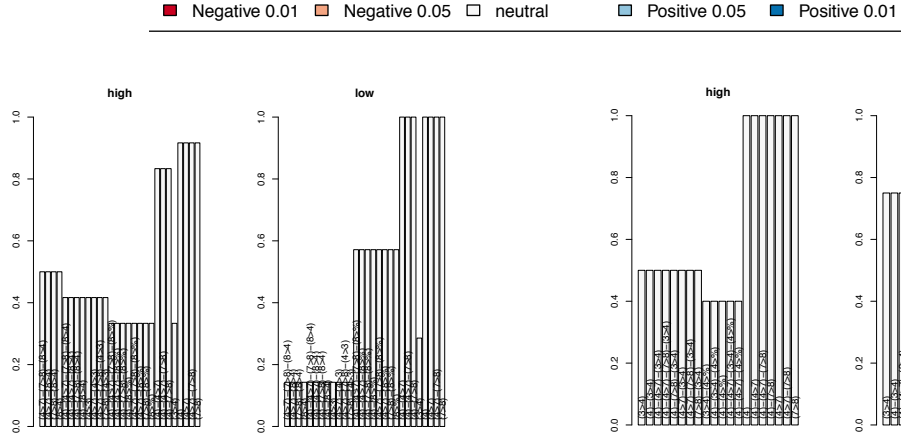


Figure 4.15 – Frequent subsequences of actions for e1 (no discrimination between groups)

Figure 4.16 – Frequent subsequences of actions for e2 (no discrimination between groups)

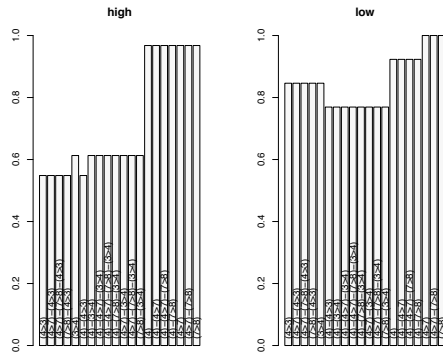


Figure 4.17 – Frequent subsequences of actions for e3 (no discrimination between groups)

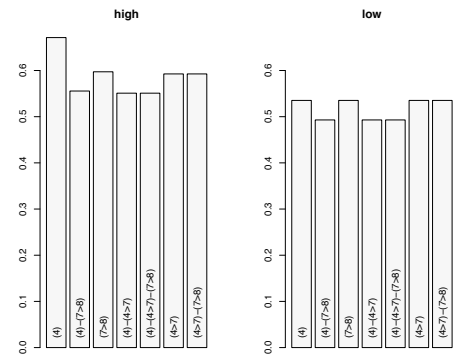


Figure 4.18 – Frequent subsequences of actions for e4 (no discrimination between groups)

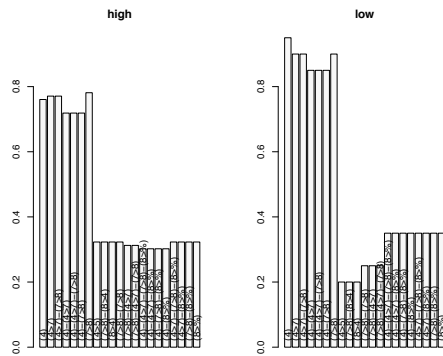


Figure 4.19 – Frequent subsequences of actions for e5 (no discrimination between groups)

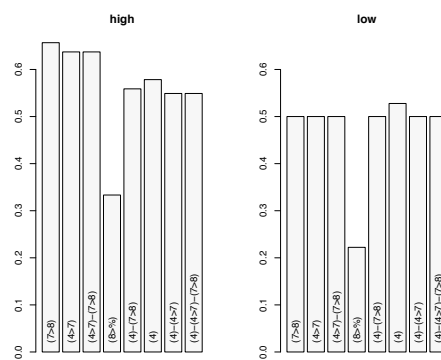


Figure 4.20 – Frequent subsequences of actions for e6 (no discrimination between groups)

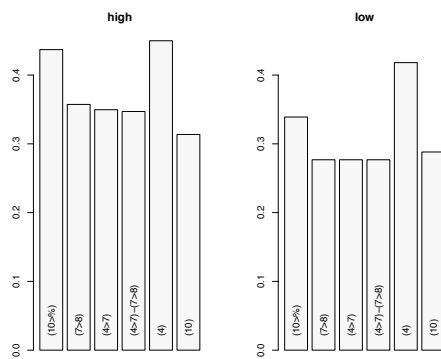


Figure 4.21 – Frequent subsequences of actions for e7 (no discrimination between groups)

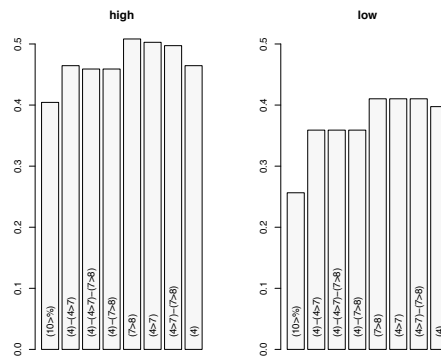


Figure 4.22 – Frequent subsequences of actions for e8 (no discrimination between groups)

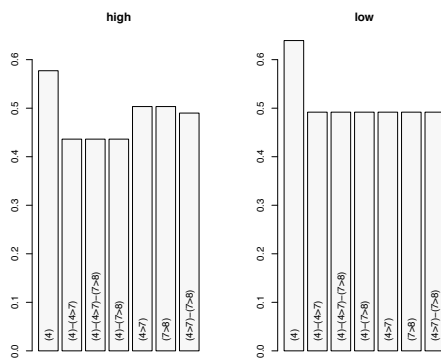


Figure 4.23 – Frequent subsequences of actions for e9 (no discrimination between groups)

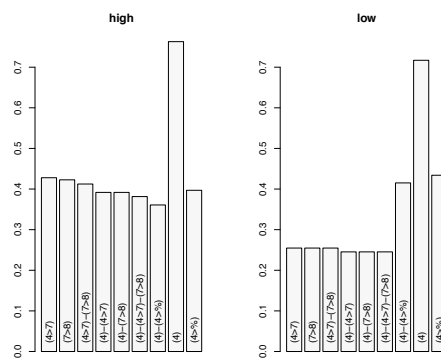


Figure 4.24 – Frequent subsequences of actions for e10 (no discrimination between groups)

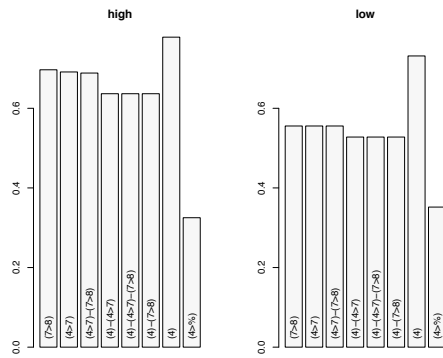


Figure 4.25 – Frequent subsequences of actions for e11 (no discrimination between groups)

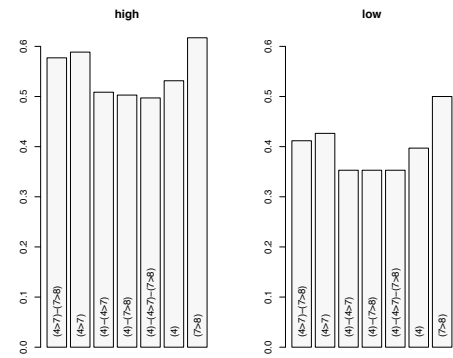


Figure 4.26 – Frequent subsequences of actions for e12 (no discrimination between groups)

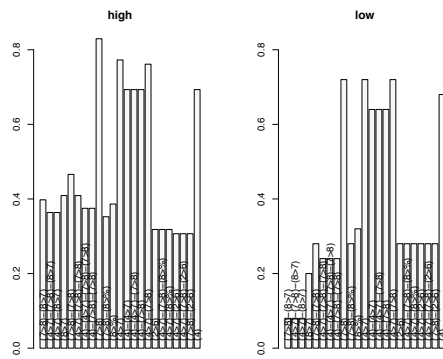


Figure 4.27 – Frequent subsequences of actions for e13 (no discrimination between groups)

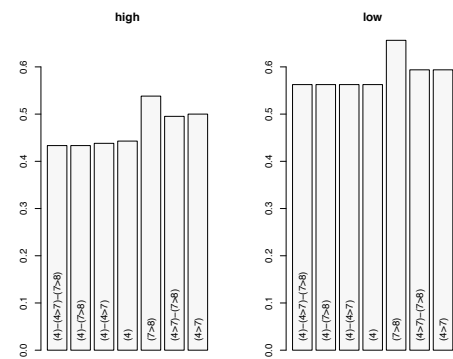


Figure 4.28 – Frequent subsequences of actions for e14 (no discrimination between groups)

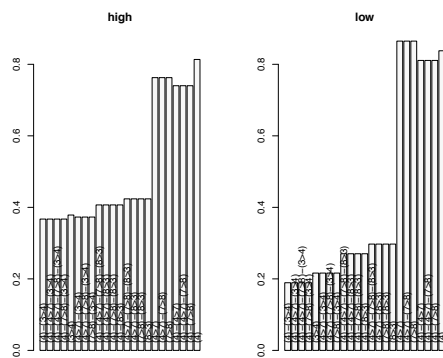


Figure 4.29 – Frequent subsequences of actions for e15 (no discrimination between groups)

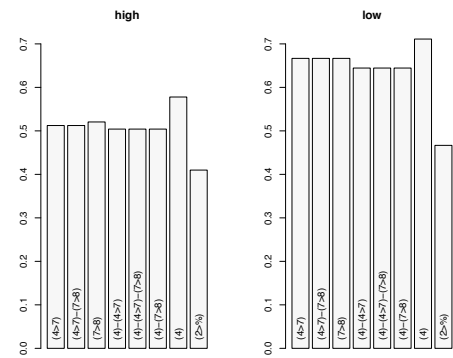


Figure 4.30 – Frequent subsequences of actions for e16 (no discrimination between groups)

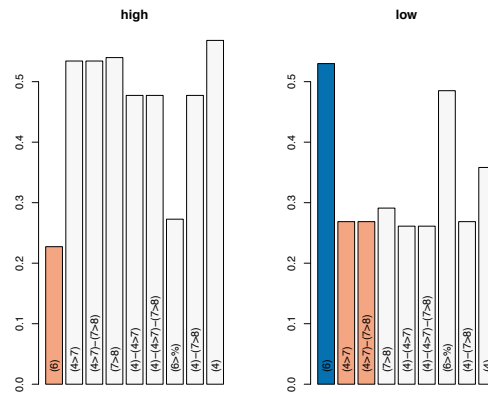


Figure 4.31 – Frequent subsequences of actions for e18 (discrimination between groups)

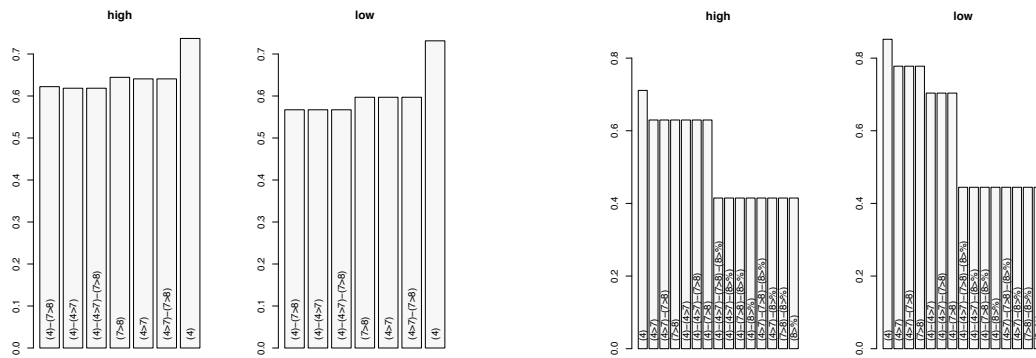


Figure 4.32 – Frequent subsequences of actions for e17 (no discrimination between groups)

Figure 4.33 – Frequent subsequences of actions for e19 (no discrimination between groups)

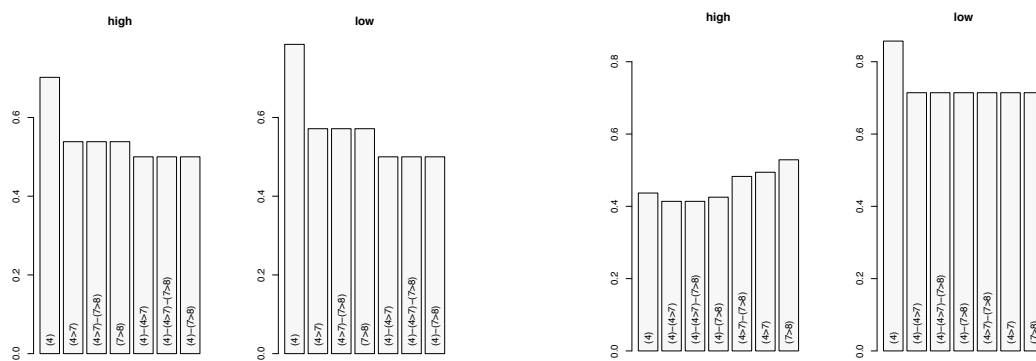


Figure 4.34 – Frequent subsequences of actions for e20 (no discrimination between groups)

Figure 4.35 – Frequent subsequences of actions for e21 (no discrimination between groups)

4.7.3 Discussion

When searching for discriminating subsequences/sequences between low and high-performing students, experiment 18 is the only experiment which showed significant results, suggesting that low and high-performing students exhibit different experimental steps for experiment 18.

Referring to what students are supposed to do in experiment 18, we know that they should find the value of the K_p (parameter number 6 in Table 4.3) using the simulation tool before starting the experiment with the CPL. Projecting on the frequency of use of parameter 6 (K_p), we see that the high-performing students changed this parameter during experiment 18 much less than low-performing students: around 20% frequency for high-performing students vs. around 50% for low-performing students (Figure 4.31). This could be an indication that high-performing student followed the suggested experimental protocol, while low-performing students did not. Another explanation emanates from the method students used to alter the mentioned parameter. On the CPL web app, students can either type in the parameter or change the value with arrows. In the case where students use the arrows to get to the desired value, each click on the arrow counts as a parameter change. So for example, if a student typed in the value 1, one event change is recorded, vs. 2 recorded events if each arrow click steps 0.5. For other discriminating subsequences ($4 > 7$ and $7 > 8$), we see that change of parameter number 4 (signal amplitude parameter), parameters 7 and 8 (T_i and T_d parameters respectively) is more frequent for high-performing students. These subsequences of steps are not instructed in the guidelines for experiment 18.

The initial goal from searching for frequent subsequences/sequence, and finding out whether they were discriminating for low and high-performing students is to identify experimental patterns, which are indicative of certain procedures: following the experimental protocol provided by the instructor, diverging from that experimental protocol and showing other behavior, etc... But given that only one experiment of the 21 in the MOOC showed significant difference between the two groups, the study was halted.

Searching for an explanation of obtaining these results which could be influenced by the procedure of building the sequences detailed in Section 4.3.3, or the choice of the *minsup* threshold discussed in Section 4.7.1, we do the following:

- Adjust the 90 seconds idleness threshold for the identification of a new sequence to 30 seconds for all experiments where students didn't have to wait for the graph to fill, and keeping it at 90 seconds for the experiments where this duration of wait is needed.
- Vary the *minsup* parameter to 0.1, 0.5 and 0.7.

All possible combinations showed similar results to the ones obtained in Section 4.7.2, suggesting that the dataset building procedure and the threshold choice were not a cause for the results.

4.8 Summary of Findings and Discussion

In this chapter, we collected and analyzed the interaction data students generated while experimenting with the electrical servo drive CPL, integrated in the Control Systems Lab MOOC.

Through the data analysis presented in Sections 4.4 and 4.5, we derived the descriptive statistics of the access to the CPLs and the experimental behavior exerted on these setups (**contribution 5**), answering **RQ5: How are student accessing and using CPLs made available to them 24/7?** The analysis of participation in Section 4.5 reveals a 77.99% participation rate in the MOOC from students enrolled in the corresponding courses. We notice that students mostly connected to the setups on the days of the scheduled lab sessions. Originating connections to the CPLs were in majority from devices not used in the lab room. As the total number of students enrolled in the MOOC is 163, and the total number of setups is 25, there was an assumption that massive access to the setups manifest, especially during the experimentation sessions. But from the analysis of queue sizes in Section 4.4.2, we see that queuing happened only 3% of the time. From the analysis of the duration spent experimenting in Section 4.4.4, and referring to the results in Table 4.4, we notice that the average time of experimenting is much bigger than the allocated time (between 6 and 11 min) for each of the modules in case of queuing. Revisiting the motivations of developing and deploying CPLs as discussed in the Introduction of this thesis, where CPLs are presented as a budgetary and logistic solution for having labs in curricula, we see that students spend less time experimenting for a lesson (6 min and 22 seconds on average) than the allocated time for the lab sessions (1 hour 30 min–2 hours). Hence, resources are utilized much less during a lab session than the allocated time for a lab session at the university, but more than the fixed time in case of queuing (allocated time per module in Table 4.4). To the extent of our knowledge, there are not similar studies to which we can compare our findings.

Through studying the effect of students' experimental behavior on their scored grades in the MOOC (**contribution 6**) in Sections 4.6 and 4.7, we answered **RQ6: How does students' experimental behavior impact their academic performance?** The statistical hypothesis tests grouping low and high-performing students showed significant effect of the total time spent experimenting throughout the course and the total number of experimental attempts. High-performing students spent more time experimenting and attempted more experiments than low-performing students. The frequent sequence mining and the statistical hypothesis testing conducted in Section 4.7 showed discriminating significant results between low and high-performing students, only for the sequences of experiment 18. Recall from Section ?? that the counts of the cited parameter could have been influenced by the method used to control it. If disregarding this possible influence on the results, they are consistent with those found in [31], where high-performing students showed more frequent use of a relative parameter to a given experiment. The results obtained of these studies are not consistent with the findings presented in [75]. The authors in this work annotated 8 types of sequences, and found significant results using ANOVA for 4 types for high-performing students, and no effect for low-performing students. It is arguable that there methodology is not correct for the data they collected, given that there was no justification for the use of a parametric method and no specification of the sample size (3 weeks of data collection).

More details are in Section 4.1.

Note: The results obtained in this chapter are for experimentation in a Control Systems course with a CPL, where there is a possibility to repeat the experiments without limitation of the resources and the reaction time of the system is quick (the maximum wait for a complete result in 1 min 30 seconds). This is not the case for all types of CPLs, for instance in the fields of biology or chemistry where there would be constraints set by the availability of material and maintenance staff. In this regard, the indications which presented or not statistically significant effects on the performance of students might not apply for other types of experiments in other fields. To the extend of our knowledge, there are no existing works in the literature for comparison.

4.9 Conclusion

In this chapter, we collected and analyzed the generated interaction data from the use of CPLs in a MOOC. We looked at two aspects of the interaction data: the descriptive statistics the access and use of the CPL infrastructure (**contribution 5**), and the effect of students' experimental behavior of their academic performance in the MOOC (**contribution 6**). We found that students mostly connect to the CPLs during lab session days, and the current resources are successfully accommodating the 163 students enrolled in the MOOC. The distributions of the total time spent experimenting throughout the course, the total number of experimental attempts and the number of parameter changes per sequence show that high-performing students tend to spend more time experimenting and attempt more experiments, while low-performing students tend to do more parameter changes per sequence. Statistically significant effects are obtained for the total time spent experimenting and the number of experiment attempts. The frequent sequence mining and corresponding hypothesis test for the experimental patterns of students showed no considerable significant results, not procuring strong evidence to pursue a deeper understanding of discriminating experimental patterns between low and high-performing students.

5 Conclusion and Outlook

In this thesis, we tackled issues pertaining to CPLs in educational settings, from the perspective of teachers, lab owners and students as stakeholders of the experience of using such systems. We first addressed challenges related to the automatic generation of web apps interfacing CPLs. Then, we addressed issues related to the collection and retrieval of the data students generate through the interaction with CPLs. And last, we tracked, analyzed and studied the impact of students' experimental behavior on their academic performance.

The automatic generation of web apps for CPLs

To enable teachers to generate web apps for CPLs, we first studied current approaches for CPL system architectures such as the ones based on the Lab as a Service paradigm. Even though they offer the possibility of individually calling a sub-group of services a CPL offers, and hence enabling the automatic generation of web apps, such paradigms do not practically support the automatic generation of user apps, since any combination of selected services is possible without any experimental meaning. Accordingly, we extended an implementation of the LaaS paradigm— the Smart Device Specification, by defining the relationships between system components through the introduction of new API calls (*configurations* corresponding to *experiments*) and data models for requests and responses exchanges. In order to validate the extension, we proposed a tool to automatically generate CPL web apps and showed its function with the Mach-Zehnder interferometer CPL. The result of this work is a solution for teachers who wish to configure CPLs according to their needs, without being tied to the availability of an app developer or lab owner. The automatic web app generator provides teachers with basic interfacing apps to integrate CPLs in online learning environments. The proposed extension has been generalized to Lab as a Service implementations of CPLs, and is included in the IEEE P1876 Standard for Networked Smart Learning Objects for Online Laboratories¹.

¹https://standards.ieee.org/email/2012_09_cfp_P1876wg_web.html

Chapter 5. Conclusion and Outlook

The research done in this part of the thesis falls under the generic titles of service discovery (searching for software services to build an application) and composition (selecting the services for the application), used by cyber-physical systems and Internet of Things communities. In our case, the services are the interface to the sensors and actuators of a CPL, and the application is the experiment conducted using the CPL. There have been efforts in both camps to enable building applications through scanning and selecting the connected sensors and actuators. Our proposal simplifies the task of application-building by specifying the possible combinations of services selection for a specific purpose (the experiments). This proposal can be further extended to describe the relationships and dependencies between services, empowering the user beyond the system-predefined combinations, and allowing the mash-up of services. In other words, instead of listing the possible *configurations* of a CPL corresponding to the *experiments*, we specify which sensors depend on which actuators and vice-versa. In the Mach-Zehnder example, we could specify that the photodiode sensors' measures are dependent on the laser function. And that the opaque beam shutters also affect the photodiode's measurements. Likewise, if the teacher (the user of the automatic user app generator) wishes to compose an *experiment* not included in the *configurations*, they can.

The collection of learner's data through interaction with CPLs

To devise an infrastructure for the collection of data generated through the use of platform-embedded CPLs, we started by eliciting the requirements of students and lab owners through questionnaires. Students are interested in their progress relative to the class, their main concern regarding data collection and retrieval is being able to control who can benefit from it, and being able to choose to deactivate tracking. Lab owners are interested in gathering data for purposes of equipment maintenance, system report and advertising. Based on the analysis of the questionnaires, we formulated the requirements for an activity tracking infrastructure composed of a vocabulary and an architectural model. We extended the xAPI specification's vocabulary in order to formalize the recording of the CPL experience, adding a new standardized way to describe a learning activity with xAPI, the CPL-xAPI experience. The proposed xAPI activity has been accepted and added to the xAPI activity registry ², the *Profiles* and *Recipes* are still under review. The proposed architectural model relies on the characteristics of the data sources to mitigate data privacy and access concerns for students and lab owners respectively. We showed how such an infrastructure can be implemented with two different CPLs: the Mach-Zehnder interferometer with an implementation on the lab owner side, and the electric servo drive with an implementation on the CPL web app side. Finally, we evaluated an implementation with students. There was a low adoption rate from the pool of students using the CPLs at the time of the study. We found that the main reason was related to system usability such as too many steps to configure and get analytics.

Implementing tracking according to the proposed infrastructure gives grounds to study a multitude of subjects in data privacy, learning analytics and human computer interaction designs. Nowadays,

²<https://goo.gl/2bFVyz>

data privacy management is drawing much attention, where a balance between preserving the privacy of users and collecting the data is still to be found. The general direction is towards enforcing certain regulations on system providers, rather than giving the users control. In our proposal, we make the user aware and allow them make an informed decision on opting in or out for tracking. Running user studies with such an infrastructure in different settings and with different pool of demographics, can help to understand the true concerns of users regarding their online privacy. In our requirements elicitation process with the group of students, we found that their main concern is not that they are tracked, but that they could be identified and that they could be able to control the sharing of their data with those who would derive value from the data. Another topic is data integration and portability. The adoption of the proposed vocabulary by the community holds a great potential in providing consistent data about the use of CPLs from different system providers, where the data is combined to provide the community with large data corpora.

The computational analysis of learners' use of CPLs

Taking advantage of MOOCs and the opportunity to work with one which integrates a CPL as a learning resource, we tracked the students of the Control Systems Lab MOOC for the academic year 2016–2017. We computationally analyzed the access and use of the labs, and we concluded that a higher participation rate does not significantly imply a better academic performance. Additionally, the time spent experimenting and the number of attempts have significant effect on the academic performance, but not the number of actions (number of parameter changes) done in an experiment. We then mined the sequences of actions students do in an experiment, in search for experimental strategies which would discriminate high and low performing students. Of the sequences mined for 21 experiments, only the sequences of one experiment showed statistically significant results.

The continued collection of similar data through the use of CPLs provides bigger datasets to better understand how students are experimenting. Note that in the study was conducted for a control systems CPL, it would be interesting to explore similarities and differences of experimental behaviors with labs for different applications, such as biology, chemistry, circuits and others. Furthermore, in this study, we analyzed the experimental activity of students outside their sequences in the MOOC (navigating through the tabs of a lesson). Adding that level of granularity of sequences to the experimental behavior can help in identifying learner types (deductive or inductive), and further understand their experimental procedures. The collected data can be further explored with advanced knowledge-mining techniques such as Deep Knowledge Tracing and variations the Hidden-Markov model which would provide a deeper understanding of students' online behavior.

A Appendix A: The directory of the xAPI statements for the running examples

In this appendix, we show the simplified xAPI statements used for the MZI and Electrical Servo Drive examples. The statements are not complete, they only contain the required fields for a valid xAPI statement and some simple to read contextualization fields. They are just to illustrate the use of the proposed extension in Chapter 3.

A.1 xAPI statments for the MZI server-side activity tracking exam-ple.

```
1 #connect to main VI
2 {
3   "actor": {
4     "name": "%s",
5     "mbox": "mailto:sd_user@nomail.com"
6   },
7   "verb": {
8     "id": "http://shindig2.epfl.ch/xapiextension.html#started"
9   },
10  "object": {
11    "id": "http://128.178.112.11"
12  },
13  "activity" : {
14    "id": "http://id.tincanapi.com/recipe/checklist/performance-observation/1",
15    "definition": {
16      "type": "http://id.tincanapi.com/activitytype/recipe",
17    },
18    "description": {
19      "en-US": "A recipe for recording the experience of CPL experimentation for
20                lab owner purposes."
21    }
22  }
```

Appendix A. Appendix A: The directory of the xAPI statements for the running examples

```
1 #stop the main VI
2 {
3     "actor": {
4         "name": "%s",
5         "mbox": "mailto:sd_administrator@nomail.com"
6     },
7     "verb": {
8         "id": "http://shindig2.epfl.ch/xapiextension.html#stopped"
9     },
10    "object": {
11        "id": "https://github.com/react-epfl/mz/blob/master/MZ_v1.0/SD_LV2014_v19/
12            SDServer_19.vi"
13    },
14    "activity" : {
15        "id": "http://id.tincanapi.com/recipe/checklist/performance-observation/1",
16        "definition": {
17            "type": "http://id.tincanapi.com/activitytype/recipe",
18        },
19        "description": {
20            "en-US": "A recipe for recording the experience of CPL experimentation for
21                lab owner purposes."
22        }
23    }
24 }
```

```
1 #disconnect from main VI
2 {
3     "actor": {
4         "name": "%s",
5         "mbox": "mailto:sd_user@nomail.com"
6     },
7     "verb": {
8         "id": "http://shindig2.epfl.ch/xapiextension.html#stopped"
9     },
10    "object": {
11        "id": "http://128.178.112.11"
12    },
13    "activity" : {
14        "id": "http://id.tincanapi.com/recipe/checklist/performance-observation/1",
15        "definition": {
16            "type": "http://id.tincanapi.com/activitytype/recipe",
17        },
18        "description": {
19            "en-US": "A recipe for recording the experience of CPL experimentation for
20                lab owner purposes."
21        }
22    }
23 }
```

```
1 #PID Real
2 {
3     "actor": {
4         "name": "%s",
5         "mbox": "mailto:sd_user@nomail.com"
6     },
7     "verb": {
8         "id": "http://shindig2.epfl.ch/xapiextension.html#used"
9     },
10    "object": {
```

A.1. xAPI statments for the MZI server-side activity tracking example.

```
11     "id": "https://github.com/react-epfl/mz/blob/master/MZ_v1.0/SD_LV2014_v19/
12     services/PID/PID_Real.vi",
13     "definition": {
14         "extensions" : {
15             "http://shindig2.epfl.ch/metadata/laser_power.html": 1
16         }
17     },
18     "activity" : {
19         "id": "http://id.tincanapi.com/recipe/checklist/performance-observation/1",
20         "definition": {
21             "type": "http://id.tincanapi.com/activitytype/recipe",
22         },
23         "description": {
24             "en-US": "A recipe for recording the experience of CPL experimentation for
25                 lab owner purposes."
26         }
27     }
28 }

1  {
2      "actor": {
3          "name": "%s",
4          "mbox": "mailto:sd_user@nomail.com"
5      },
6      "verb": {
7          "id": "http://shindig2.epfl.ch/xapiextension.html#used"
8      },
9      "object": {
10         "id": "https://github.com/react-epfl/mz/blob/master/MZ_v1.0/SD_LV2014_v19/
11         services/PID/PID_Real.vi",
12         "definition": {
13             "extensions" : {
14                 "http://shindig2.epfl.ch/metadata/photodiode.html": 1
15             }
16         },
17         "activity" : {
18             "id": "http://id.tincanapi.com/recipe/checklist/performance-observation/1",
19             "definition": {
20                 "type": "http://id.tincanapi.com/activitytype/recipe",
21             },
22             "description": {
23                 "en-US": "A recipe for recording the experience of CPL experimentation for
24                     lab owner purposes."
25             }
26         }
27     }
28 }

1  #PWM1
2  {
3      "actor": {
4          "name": "%s",
5          "mbox": "mailto:sd_user@nomail.com"
6      },
7      "verb": {
8          "id": "http://shindig2.epfl.ch/xapiextension.html#used"
9      },
10     "object": {
```

Appendix A. Appendix A: The directory of the xAPI statements for the running examples

```
11     "id": "https://github.com/react-epfl/mz/blob/master/MZ_v1.0/SD_LV2014_v19/
12     services/PWM1/PWM1_Real.vi",
13     "definition": {
14         "extensions" : {
15             "http://shindig2.epfl.ch/metadata/beam_splitter0.html": 1
16         }
17     },
18     "activity" : {
19         "id": "http://id.tincanapi.com/recipe/checklist/performance-observation/1",
20         "definition": {
21             "type": "http://id.tincanapi.com/activitytype/recipe",
22         },
23         "description": {
24             "en-US": "A recipe for recording the experience of CPL experimentation for
25             lab owner purposes."
26         }
27     }
28 }

1 #PWM2
2 {
3     "actor": {
4         "name": "%s",
5         "mbox": "mailto:sd_user@nomail.com"
6     },
7     "verb": {
8         "id": "http://shindig2.epfl.ch/xapiextension.html#used"
9     },
10    "object": {
11        "id": "https://github.com/react-epfl/mz/blob/master/MZ_v1.0/SD_LV2014_v19/
12        services/PWM2/PWM2_Real.vi",
13        "definition": {
14            "extensions" : {
15                "http://shindig2.epfl.ch/metadata/beam_splitter1.html": 1
16            }
17        },
18        "activity" : {
19            "id": "http://id.tincanapi.com/recipe/checklist/performance-observation/1",
20            "definition": {
21                "type": "http://id.tincanapi.com/activitytype/recipe",
22            },
23            "description": {
24                "en-US": "A recipe for recording the experience of CPL experimentation for
25                lab owner purposes."
26            }
27        }
28    }
29 }

1 #Piezo
2 {
3     "actor": {
4         "name": "%s",
5         "mbox": "mailto:sd_user@nomail.com"
6     },
7     "verb": {
8         "id": "http://shindig2.epfl.ch/xapiextension.html#used"
9     },
10    "object": {
```

A.2. xAPI statments for the Servo Drive Client-App activity tracking example.

```
11     "id": "https://github.com/react-epfl/mz/blob/master/MZ_v1.0/SD_LV2014_v19/
    services/Piezo/Piezo_Real.vi",
12     "definition": {
13         "extensions" : {
14             "http://shindig2.epfl.ch/metadata/piezo_actuator.html": 1
15         }
16     },
17     "activity" : {
18         "id": "http://id.tincanapi.com/recipe/checklist/performance-observation/1",
19         "definition": {
20             "type": "http://id.tincanapi.com/activitytype/recipe",
21         },
22         "description": {
23             "en-US": "A recipe for recording the experience of CPL experimentation for
24                 lab owner purposes."
25         }
26     }
27 }
```

A.2 xAPI statments for the Servo Drive Client-App activity tracking example.

```
1 #configure an actuator
2 "actor": {
3     "mbox": "mailto:" + uID + "@courseware.epfl.ch"
4 },
5 "verb": {
6     "id": "http://adlnet.gov/expapi/verbs/configured"
7 },
8 "object": {
9     "id": "http://" + baseUrl
10 },
11 "context": {
12     "extensions": {
13         "http://baseUrl/configurationId": "'" + ExpID + "'"
14     },
15     "activity" : {
16         "id": "http://id.tincanapi.com/recipe/checklist/performance-observation/1",
17         "definition": {
18             "type": "http://id.tincanapi.com/activitytype/recipe",
19         },
20         "description": {
21             "en-US": "A recipe for recording the experience of CPL experimentation for
22                 students purposes."
23         }
24     }
25 }

1 #suspend the experiment
2 "actor": {
3     "mbox": "mailto:" + uID + "@courseware.epfl.ch",
4     "objectType": "Agent"
5 },
6 "verb": {
7     "id": "http://adlnet.gov/expapi/verbs/suspended"
8 },
9 "object": {
```

Appendix A. Appendix A: The directory of the xAPI statements for the running examples

```
10         "id": "http://" + baseUrl
11     },
12     "context": {
13         "extensions": {
14             "http://baseUrl/configurationId": "'" + ExpID + "'"
15         },
16     "activity" : {
17         "id": "http://id.tincanapi.com/recipe/checklist/performance-observation/1",
18         "definition": {
19             "type": "http://id.tincanapi.com/activitytype/recipe",
20         },
21         "description": {
22             "en-US": "A recipe for recording the experience of CPL experimentation for
23                 students purposes."
24         }
25     }
26 }

1 #save the experimental results
2 "actor": {
3     "mbox": "mailto:" + uID + "@courseware.epfl.ch"
4 },
5 "verb": {
6     "id": "http://adlnet.gov/expapi/verbs/saved"
7 },
8 "object": {
9     "id": "http://" + baseUrl
10 },
11 "context": {
12     "extensions": {
13         "http://baseUrl/configurationId": "'" + ExpID + "'"
14     }
15 },
16 "activity" : {
17     "id": "http://id.tincanapi.com/recipe/checklist/performance-observation/1",
18     "definition": {
19         "type": "http://id.tincanapi.com/activitytype/recipe",
20     },
21     "description": {
22         "en-US": "A recipe for recording the experience of CPL experimentation for
23             students purposes."
24     }
25 }
26 }

1 #start the experiment
2 "actor": {
3     "mbox": "mailto:" + uID + "@courseware.epfl.ch",
4     "objectType": "Agent"
5 },
6 "verb": {
7     "id": "http://activitystrea.ms/schema/1.0/started"
8 },
9 "object": {
10     "id": "http://" + baseUrl
11 },
12 "context": {
13     "extensions": {
14         "http://baseUrl/configurationId": "'" + ExpID + "'"
15     }
16 },
17 "activity" : {
18     "id": "http://id.tincanapi.com/recipe/checklist/performance-observation/1",
```

A.2. xAPI statments for the Servo Drive Client-App activity tracking example.

```
19     "definition": {
20         "type": "http://id.tincanapi.com/activitytype/recipe",
21     },
22     "description": {
23         "en-US": "A recipe for recording the experience of CPL experimentation for
24             students purposes."
25     }
26 }
27
28 1 #resume the experiment
29 2 "actor": {
30     3     "mbox": "mailto:" + uID + "@courseware.epfl.ch"
31     4 },
32     5     "verb": {
33     6         "id": "http://adlnet.gov/expapi/verbs/resumed"
34     7     },
35     8     "object": {
36     9         "id": "http://" + baseUrl
37     10    },
38    11    "context": {
39    12        "extensions": {
40    13            "http://baseUrl/configurationId": "'" + ExpID + "'"
41    14        }
42    15    },
43    16    "activity" : {
44    17        "id": "http://id.tincanapi.com/recipe/checklist/performance-observation/1",
45    18        "definition": {
46    19            "type": "http://id.tincanapi.com/activitytype/recipe",
47    20        },
48    21        "description": {
49    22            "en-US": "A recipe for recording the experience of CPL experimentation for
50                students purposes."
51    23        }
52    24    }
53 }
54
55 1 #terminate the experiment
56 2 "actor": {
57     3     "mbox": "mailto:" + uID + "@courseware.epfl.ch",
58     4     "objectType": "Agent"
59     5 },
60     6     "verb": {
61     7         "id": "http://adlnet.gov/expapi/verbs/terminated"
62     8     },
63     9     "object": {
64    10         "id": "http://" + baseUrl
65    11     },
66    12     "context": {
67    13         "extensions": {
68    14             "http://baseUrl/configurationId": ExpID
69    15         }
70    16     },
71    17     "activity" : {
72    18         "id": "http://id.tincanapi.com/recipe/checklist/performance-observation/1",
73    19         "definition": {
74    20             "type": "http://id.tincanapi.com/activitytype/recipe",
75    21         },
76    22         "description": {
77    23             "en-US": "A recipe for recording the experience of CPL experimentation for
78                 students purposes."
79    24         }
80    25     }
81 }
```


Bibliography

- [1] Vincent Aleven, Jonathan Sewall, Octav Popescu, Franceska Xhakaj, Dhruv Chand, Ryan Baker, Yuan Wang, George Siemens, Carolyn Rosé, and Dragan Gasevic. The beginning of a beautiful friendship? intelligent tutoring systems and moocs. In *International Conference on Artificial Intelligence in Education*, pages 525–528. Springer, 2015.
- [2] Evgeny Bogdanov, Freddy Limpens, Na Li, Sandy El Helou, Christophe Salzmann, and Denis Gillet. A social media platform in higher education. In *Global engineering education conference (educon), 2012 ieee*, pages 1–8. IEEE, 2012.
- [3] D Boyd and A Marwick. Social privacy in networked publics: teens’ attitudes, practices, and strategies’, paper given at, 2011.
- [4] MJ. Callaghan, J. Harkin, TM. McGinnity, and LP. Maguire. Cost effectiveness issues in remote experimentation. *2004 IEEE International Conference on Systems, Man and Cybernetics*, 5:4700–4704, 2004.
- [5] Richard Carr. Gang of four design patterns.
- [6] Jesús Chacón, Gonzalo Farias, Hector Vargas, Antonio Visioli, and Sebastián Dormido. Remote interoperability protocol: A bridge between interactive interfaces and engineering systems** this work has been funded by the national plan project dpi2012-31303 of the spanish ministry of science and innovation and feder funds. *IFAC-PapersOnLine*, 48(29):247–252, 2015.
- [7] Jesus Chacon, Hector Vargas, Gonzalo Farias, Jose Sanchez, and Sebastian Dormido. Ejs, jil server, and labview: An architecture for rapid development of remote labs. *IEEE Transactions on Learning Technologies*, 8(4):393–401, 2015.
- [8] M Bishop Christopher. *PATTERN RECOGNITION AND MACHINE LEARNING*.
- [9] Stephanie Chuffart-Finterwald. Data protection in switzerland: overview.
- [10] Roger Clarke. Privacy impact assessment: Its origins and development. *Computer law & security review*, 25(2):123–135, 2009.
- [11] IMS Global Learning Consortium. Ims caliper analytics implementation guide.

Bibliography

- [12] The Swiss Federal Council. Federal act on data protection (fadp).
- [13] EPFL courseware. Politique de confidentialité.
- [14] Ryan SJ d Baker, Albert T Corbett, and Vincent Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *International Conference on Intelligent Tutoring Systems*, pages 406–415. Springer, 2008.
- [15] Balakrishnan Dasarathy, Kevin Sullivan, Douglas C. Schmidt, Douglas H. Fisher, and Adam Porter. The past, present, and future of moocs and their relevance to software engineering. *FOSE 2014 Proceedixngs of the on Future of Software Engineering*, pages 212–224, 2014.
- [16] Dan Davis, Guanliang Chen, Claudia Hauff, and Geert-Jan Houben. Gauging mooc learners’ adherence to the designed learning path. In *EDM*, pages 54–61, 2016.
- [17] Ton de Jong, Sofoklis Sotiriou, and Denis Gillet. Innovations in stem education: the go-lab federation of online labs. *Smart Learning Environments*, 1(1):3, 2014.
- [18] Luis de la Torre, José Sánchez, Tiago Faustino Andrade, and Maria Teresa Restivo. Easy creation and deployment of javascript remote labs with ejss and moodle. In *2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, pages 260–261. IEEE, 2016.
- [19] Ángel Del Blanco, Ángel Serrano, Manuel Freire, Iván Martínez-Ortiz, and Baltasar Fernández-Manjón. E-learning standards and learning analytics. can data collection be improved by using standard data models? In *Global Engineering Education Conference (EDUCON), 2013 IEEE*, pages 1255–1261. IEEE, 2013.
- [20] Pierre Dillenbourg, Guillaume Zufferey, Hamed Alavi, Patrick Jermann, Son Do-Lenh, Quentin Bonnard, Sébastien Cuendet, and Frédéric Kaplan. Classroom orchestration: The third circle of usability. *CSCL2011 Proceedings*, 1:510–517, 2011.
- [21] A Downes, A Shahrazad, and R Smith. Sharing between lrss: a collaborative experiment in practical interoperability, 2015.
- [22] Yucong Duan, Guohua Fu, Nianjun Zhou, Xiaobing Sun, Nanjangud C Narendra, and Bo Hu. Everything as a service (xaas) on the cloud: Origins, current and future trends. In *Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on*, pages 621–628. IEEE, 2015.
- [23] Erik Duval. Attention please!: learning analytics for visualization and recommendation. In *Proceedings of the 1st International Conference on Learning Analytics and Knowledge*, pages 9–17. ACM, 2011.
- [24] EduApps. Writing lti stuff.
- [25] RP Photonics Encyclopedia. Beam splitters.

-
- [26] Hamide Ertepinar and Ömer Geban. Effect of instruction supplied with the investigative-oriented laboratory approach on achievement in a science course. *Educational Research*, 38(3):333–341, 1996.
- [27] Tessa HS Eysink, Ton de Jong, Kirsten Berthold, Bas Kolloffel, Maria Opfermann, and Pieter Wouters. Learner performance in multimedia learning arrangements: An analysis across instructional approaches. 2009.
- [28] Louis Pierre Faucon. Exploring students learning approaches in moocs. Technical report, 2017.
- [29] Richard M Felder, Linda K Silverman, et al. Learning and teaching styles in engineering education. *Engineering education*, 78(7):674–681, 1988.
- [30] Lauren Fratamico, Cristina Conati, Samad Kardan, and Ido Roll. Applying a framework for student modeling in exploratory learning environments: Comparing data representation granularity to handle environment complexity. *International Journal of Artificial Intelligence in Education*, 27(2):320–352, 2017.
- [31] Lauren Fratamico, Sarah Perez, and Ido Roll. A visual approach towards knowledge engineering and understanding how students learn in complex environments. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*, pages 13–22. ACM, 2017.
- [32] Chase Geigle and ChengXiang Zhai. Modeling mooc student behavior with two-layer hidden markov models. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*, pages 205–208. ACM, 2017.
- [33] Andrew Gelman. Commentary: P values and statistical practice. *Epidemiology*, 24(1):69–72, 2013.
- [34] Denis Gillet, Ton De Jong, Sofoklis Sotirou, and Christophe Salzmann. Personalised learning spaces and federated online labs for stem education at school. In *Global Engineering Education Conference (EDUCON), 2013 IEEE*, pages 769–773. IEEE, 2013.
- [35] Denis Gillet, Andrii Vozniuk, Maria Jesus Rodriguez Triana, and Adrian Christian Holzer. Agile, versatile, and comprehensive social media platform for creating, sharing, exploiting, and archiving personal learning spaces, artifacts, and traces. In *The World Engineering Education Forum*, number EPFL-CONF-221529, 2016.
- [36] Sten Govaerts, Yiwei Cao, Andrii Vozniuk, Adrian Holzer, Danilo Garbi Zutin, Elio San Cristóbal Ruiz, Lars Bollen, Sven Manske, Nils Faltin, Christophe Salzmann, et al. Towards an online lab portal for inquiry-based stem learning at school. In *International Conference on Web-Based Learning*, pages 244–253. Springer, 2013.
- [37] Sander Greenland and Charles Poole. Living with p values: resurrecting a bayesian perspective on frequentist statistics. *Epidemiology*, 24(1):62–68, 2013.

Bibliography

- [38] Elaine Harger. Lessig, lawrence. code and other laws of cyberspace. new york: Basic bks., 1999. 297p. 30, *cloth*(isbn046503912x);14, paper (isbn 0465039138). lc 00-271772. *College & Research Libraries*, 62(1):94–97, 2001.
- [39] V Judson Harward, Jesus A Del Alamo, Steven R Lerman, Philip H Bailey, Joel Carpenter, Kimberley DeLong, Chris Felknor, James Hardison, Bryant Harrison, Imad Jabbour, et al. The ilab shared architecture: A web services infrastructure to build communities of internet accessible laboratories. *Proceedings of the IEEE*, 96(6):931–950, 2008.
- [40] Tobias Hecking, Sven Manske, Lars Bollen, Sten Govaerts, Andrii Vozniuk, and H Ulrich Hoppe. A flexible and extendable learning analytics infrastructure. In *International Conference on Web-Based Learning*, pages 123–132. Springer, 2014.
- [41] Peter High. Lessons from the ceo of the first ever mooc. *Forbes*. <http://www.forbes.com/sites/peterhigh/2013/12/31/lessons-from-the-ceo-of-the-first-ever-mooc>, 2013.
- [42] Avi Hofstein and Vincent N Lunetta. The laboratory in science education: Foundations for the twenty-first century. *Science education*, 88(1):28–54, 2004.
- [43] Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, and David Boyle. *From Machine-to-machine to the Internet of Things: Introduction to a New Age of Intelligence*. Academic Press, 2014.
- [44] Liang Hu, Nannan Xie, Zhejun Kuang, and Kuo Zhao. Review of cyber-physical system architecture. In *Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (ISORCW), 2012 15th IEEE International Symposium on*, pages 25–30. IEEE, 2012.
- [45] René F Kizilcec, Mar Pérez-Sanagustín, and Jorge J Maldonado. Self-regulated learning strategies predict learner behavior and goal attainment in massive open online courses. *Computers & education*, 104:18–33, 2017.
- [46] William H Kruskal and W Allen Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association*, 47(260):583–621, 1952.
- [47] Advance Distributed Learning. xapi teachnical specifications.
- [48] Daniel Lowe. Mools: Massive open online laboratories: An analysis of scale and feasibility. In *Remote Engineering and Virtual Instrumentation (REV), 2014 11th International Conference on*, pages 1–6. IEEE, 2014.
- [49] Moodle. External tool settings.
- [50] Shinichi Nakagawa and Innes C Cuthill. Effect size, confidence interval and statistical significance: a practical guide for biologists. *Biological reviews*, 82(4):591–605, 2007.

-
- [51] Anh Vu Nguyen Ngoc. *WEB-BASED INTERACTION AND COLLABORATION IN FLEXIBLE ENGINEERING EDUCATION. AN ARTIFACT-BASED APPROACH*. PhD thesis, ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, 2006.
- [52] Tim O'Reilly. Web 2.0: Compact definition?
- [53] Zachary A Pardos, Steven Tang, Daniel Davis, and Christopher Vu Le. Enabling real-time adaptivity in moocs with a personalized next-step recommendation framework. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*, pages 23–32. ACM, 2017.
- [54] Judea Pearl. Bayesianism and causality, or, why i am only a half-bayesian. In *Foundations of bayesianism*, pages 19–36. Springer, 2001.
- [55] Margus Pedaste, Mario Mäeots, Leo A Siiman, Ton De Jong, Siswa AN Van Riesen, Ellen T Kamp, Constantinos C Manoli, Zacharias C Zacharia, and Eleftheria Tsourlidaki. Phases of inquiry-based learning: Definitions and the inquiry cycle. *Educational research review*, 14:47–61, 2015.
- [56] Alexsandro Pereira, Fernanda Ostermann, and Cláudio Cavalcanti. On the use of a virtual mach–zehnder interferometer in the teaching of quantum mechanics. *Physics Education*, 44(3):281, 2009.
- [57] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. Deep knowledge tracing. In *Advances in Neural Information Processing Systems*, pages 505–513, 2015.
- [58] Azhar Syed Rauf, Mohd Fadzil Daud, and Mohd Nihra Haruzuan Bin Mohamad Said. Moocs in engineering education – a literature review. *Regional Conference in Engineering Education 2016 - Malaysia*, 2016.
- [59] Ch Salzmann and Denis Gillet. Remote labs and social media: Agile aggregation and exploitation in higher engineering education. In *2011 IEEE Global Engineering Education Conference (EDUCON)*, pages 307–311. IEEE, 2011.
- [60] Christophe Salzmann, , Wissam Halimi, Denis Gillet, and Sten Govaerts. Deploying large scale online labs with smart devices. *Cyber-Physical Laboratories in Engineering and Science Education*, page (In press), 2018.
- [61] Christophe Salzmann and Denis Gillet. Smart device paradigm standardization for online labs. In *4th IEEE Global Engineering Education Conference (EDUCON)*, 2013.
- [62] Christophe Salzmann, Denis Gillet, and Yves Piguet. Mools for moocs: A first edx scalable implementation. In *Remote Engineering and Virtual Instrumentation (REV), 2016 13th International Conference on*, pages 246–251. IEEE, 2016.

Bibliography

- [63] Christophe Salzmänn, Sten Govaerts, Wissam Halimi, and Denis Gillet. The smart device specification for remote labs. In *Remote engineering and virtual instrumentation (rev), 2015 12th international conference on*, pages 199–208. IEEE, 2015.
- [64] Christophe Salzmänn. User manual for online tool - controls system’ hand on sessions. Course Manual.
- [65] Thomas Scofield. Basic anova concepts. University Lecture Notes.
- [66] Mina Shirvani Boroujeni and Pierre Dillenbourg. Discovery and temporal analysis of latent study patterns from mooc interaction sequences. In *8th International Learning Analytics and Knowledge Conference (LAK18)*, number EPFL-CONF-232491, 2018.
- [67] George Siemens and Ryan SJ d Baker. Learning analytics and educational data mining: towards communication and collaboration. In *Proceedings of the 2nd international conference on learning analytics and knowledge*, pages 252–254. ACM, 2012.
- [68] Ian Sommerville and Pete Sawyer. *Requirements engineering: a good practice guide*. John Wiley & Sons, Inc., 1997.
- [69] Ying Tan, Steve Goddard, and Lance C Perez. A prototype architecture for cyber-physical systems. *ACM Sigbed Review*, 5(1):26, 2008.
- [70] Mohamed Tawfik, Christophe Salzmänn, Denis Gillet, David Lowe, Hamadou Saliah-Hassane, Elio Sancristobal, and Manuel Castro. Laboratory as a service (laas): A model for developing and implementing remote laboratories as modular components. In *Remote Engineering and Virtual Instrumentation (REV), 2014 11th International Conference on*, pages 11–20. IEEE, 2014.
- [71] TinCan/xAPI. Recipes: How it works.
- [72] TinCan/xAPI. Statement design.
- [73] Ll Tobarra, Salvador Ros, Rafael Pastor, Roberto Hernández, Manuel Castro, Abdullah Al-Zoubi, Mamoun Dmour, Antonio Robles-Gomez, AC Caminero, and Jesús Cano. An integrated example of laboratories as a service into learning management systems. *International Journal of Online Engineering*, 12(9), 2016.
- [74] UNESCO. What are Open Educational Resources (OER). <http://www.unesco.org/new/en/communication-and-information/access-to-knowledge/open-educational-resources/what-are-open-educational-resources-oers/>, 2016. [Online; accessed 30-December-2016].
- [75] Rémi Venant, Kshitij Sharma, Pierre Dillenbourg, Philippe Vidal, and Julien Broisin. A study of learners’ behaviors in hands-on learning situations and their correlation with academic performance. In *International Conference on Artificial Intelligence in Education*, pages 570–573. Springer, 2017.

- [76] Andrii Vozniuk. *Enhancing Social Media Platforms for Educational and Humanitarian Knowledge Sharing: Analytics, Privacy, Discovery, and Delivery Aspects*. PhD thesis, ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, 2017.
- [77] Andrii Vozniuk, Sten Govaerts, Lars Bollen, Sven Manske, Tobias Hecking, and Denis Gillet. Angela: Putting the teacher in control of student privacy in the online classroom. In *Information Technology Based Higher Education and Training (ITHET)*, 2014, pages 1–4. IEEE, 2014.
- [78] Ronald E Walpole, Raymond H Myers, Sharon L Myers, and Keying Ye. *Probability and statistics for engineers and scientists*, volume 5. Macmillan New York, 1993.
- [79] Wikipedia. Alison (company).
- [80] Wikipedia. Application programming interface.
- [81] Chris Wild and George Seber. The wilcoxon rank-sum test, 2011.
- [82] David Wright. The state of the art in privacy impact assessment. *Computer Law & Security Review*, 28(1):54–61, 2012.
- [83] Li Yongfu, Sun Dihua, Liu Weining, and Zhang Xuebo. A service-oriented architecture for the transportation cyber-physical systems. In *Control Conference (CCC), 2012 31st Chinese*, pages 7674–7678. IEEE, 2012.
- [84] Michael V Yudelson, Kenneth R Koedinger, and Geoffrey J Gordon. Individualized bayesian knowledge tracing models. In *International Conference on Artificial Intelligence in Education*, pages 171–180. Springer, 2013.
- [85] K P Zetie, S F Adams, and R M Tocknell. How does a mach-zehnder interferometer work? *Physics Education*, 35(1):46, 2000.
- [86] Danilo G Zutin, Michael Auer, Pablo Ordu, Christian Kreiter, et al. Online lab infrastructure as a service: A new paradigm to simplify the development and deployment of online labs. In *2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, pages 208–214. IEEE, 2016.



List of Abbreviations

- API: Application Programming Interface
- App: Application
- CPL: Cyber-Physical Lab
- CPS: Cyber-Physical System
- ICT: Information and Communication Technologies
- ILS: Inquiry Learning Space
- IoT: Internet of Things
- Lab: Laboratory
- LMS: Learning Management System
- LO: Lab owner
- LRS: Learning Record Store
- LTI: Learning Tools Interoperability
- MOOC: Massive Open Online Course
- MZI: Mach-Zehnder Interferometer
- OER: Open Educational Resource
- OEL: Open Education Lab
- RL: Remote Laboratory
- SD: Smart Device
- STEM: Science, Technology, Engineering, and Mathematics
- SOA: Service Oriented Architecture
- SOC: Service Oriented Computing
- UI: User Interface
- xAPI: Experience API

Wissam Halimi

✉ wissam.halimi@epfl.ch • www.linkedin.com/in/wissamhalimi

Education

EPFL

Lausanne - Switzerland

PhD in Electrical Engineering

2013–2018

PhD Thesis: Topics in Cyber-Physical Labs: Configurations, Data Collection and Analysis.

Lebanese American University

Byblos - Lebanon

Bachelor of Engineering in Computer Engineering

2007–2013

Capstone Project: Real-time Video Cartoonification with FPGAs.

Ecole Maronite de la Sainte Famille

Tripoli - Lebanon

General Sciences Bachalaureate - Math and Physics

2007

Projects

Go-Lab: Part of the technical cluster of this EU funded project. Gathered requirements from teachers and cyber-physical labs providers, implemented web-based educational applications, developed template-based solutions for two target embedded devices: myRIO of National Instruments and the BeagleBone Black, maintained a local copy of the Apache Shindig server, developed new APIs and libraries, contributed and led deliverable submissions.

SCOPES: Funded by the Swiss National Science Foundation to transfer in-house developed technology to eastern European countries. Was responsible for introducing our academic partners to the Go-Lab solutions for the development and deployment CPLs. Also advised them on best software and hardware solutions for their purposes.

IEEE P1876: Been actively participating for the last 3 years in the writing of the standard, as a member of the IEEE P1876 Working Group for Networked Smart Learning Objects for Online Laboratories. Participated in face-to-face meetings and bi-weekly virtual meetings. Responsible for the modelling of cyber-physical labs as configurable software services and defining the integration requirements in web-based platforms.

Languages

Arabic: Native

French: Full professional proficiency

Instruction language at school

English: Full professional proficiency

Instruction language at university

Skills

Programming: LabVIEW, Java, C++, Javascript, Matlab, Python and R (for data analysis)

Data visualization tools: D3js, NVD3, Highcharts, Plotly, Tableau (Beginner), Circos

Databases: SQL, MongoDB

Web development frameworks: MEAN stack, Ruby on Rails

Version control : Github, SVN

Professional affiliations

IEEE: Member since 2008

ACM: Member since 2010

List of Publications

- Halimi, W., Salzmänn, C. and Gillet, D., 2017, April. Access to Massive Open Online Labs through a MOOC. In Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale (pp. 331-334). ACM.
- Halimi, W., Salzmänn, C. and Gillet, D. 2017. An Activity Tracking Infrastructure for Embedded Open Educational Labs. Supporting the Needs of Lab Owners and Students. In 20th International Conference on Interactive Collaborative Learning. In Lecture Notes in Computer Science, Springer.
- Salzmänn, C., Halimi, Wissam., Gillet, Denis. and Govaerts, Sten., 2018. Deploying Large Scale Online Labs with Smart Devices. In Cyber-Physical Laboratories in Engineering and Science Education. Springer (In press).
- Halimi, W., Salzmänn, C., Gillet, D. and Saliah-Hassane, H., 2018. Standardization Layers for Remote Laboratories as Services and Open Educational Resources. In Online Engineering & Internet of Things (pp. 874-884). Springer, Cham.
- Halimi, W., Salzmänn, C., Jamkojian, H. and Gillet, D., 2018. Enabling the automatic generation of user interfaces for remote laboratories. In Online Engineering & Internet of Things (pp. 778-793). Springer, Cham.
- Halimi, W., Salzmänn, C. and Gillet, D., 2016, February. The Mach-Zehnder Interferometer? A smart remote experiment based on a software template. In Remote Engineering and Virtual Instrumentation (REV), 2016 13th International Conference on (pp. 287-292). IEEE
- Halimi, W., Salzmänn, C., Gillet, D. and , 2015, June. The Smart Wind Turbine Lab. In 3rd Experiment@ International Conference (exp. at'15).
- Rodriguez-Triana, M.J., Govaerts, S., Halimi, W., Holzer, A., Salzmänn, C., Vozniuk, A., de Jong, T., Sotirou, S. and Gillet, D., 2014, November. Rich open educational resources for personal and inquiry learning: Agile creation, sharing and reuse in educational social media platforms. In Web and Open Access to Learning (ICWOAL), 2014 International Conference on (pp. 1-6). IEEE.
- Salzmänn, C., Govaerts, S., Halimi, W. and Gillet, D., 2015, February. The smart device specification for remote labs. In Remote engineering and virtual instrumentation (rev), 2015 12th international conference on (pp. 199-208). IEEE.

